Development of a pattern classification method (LDA) to improve signal selection and cuts optimization

Julien Faivre

22 septembre 2005
Star note #SN0448

Abstract: As soon as the signal of the analysed particle is entangled with orders of magnitude more background, its analysis benefits from the use of a pattern classification method to discriminate the signal out of the background candidates, with an improvement that rises when the number of variables used is increased. The method that we present here is the basic Linear Discriminant Analysis (LDA) and its modifications : two improvements made necessary by the extreme signal-to-background conditions encountered in our field: the use of cascaded cuts and of a locally optimized criterion.

We present the various algorithms, the way of use and tips, as well as an example and results obtained for the multi-strange baryons Ξ and Ω in STAR's 200 *GeV* `Au-Au` year 2001 data. We show that optimized multicut LDA has a higher performance than classical cuts, provides a very fast and easy cut optimization, can be used to estimate a systematic error due to the cuts, and allows for an automatic and optimal use of the inner tracking layers in the cuts.

Introduction

This note describes the adaptation of Linear Discriminant Analysis (LDA), a pattern classification method widely used in data processing, to the conditions of a strange baryon analysis in our field, i.e. to the extraction of a small amount of signal out of an overwhelming background. Some developments were necessary due to the difference in statistics between the signal and the background.

The small production yield of the searched particles and the so far limited data statistics indeed made it necessary to optimize the cuts that select the signal out of the background. The development of a new selection method aimed both at raising the final signal statistics in a fast and easy-to-use way, and at simplifying the cuts optimization by transforming the cuts space into a monodimensional sub-space, which is impossible to do in a classical analysis. In this note, the terms “classical analysis” or “classical cuts” will be employed for a selection based on cutting all the observables separately, by a “steep cut”.

The resulting method, the optimized multicut LDA, can be used in any situation of our field where the signal of the particle searched is drowned in a background that is several orders of magnitude higher, as soon as two or more variables are available for cutting.

The first section of this note is a brief introduction to the field of pattern classification. In the second section, examples of usable variables are given. The third section explains where signal and background candidates can be taken from to train a supervised method. Then, sections four and five explain how Fisher-LDA is working and what are the improvements brought to this basis to build a method that is usable in our environment. The LDA cut-tuning “in practice” is explained in the sixth section, while the last one shows the results obtained on the Ξ and Ω multi-strange baryons on the 200 *GeV* `Au-Au` year2 dataset.

Paragraphs 1 and 4.2 have been written with the great help of [1, 2]. Paragraph 4.2 is also inspired from [3]. [2] has also been useful for paragraph 5.5.4.

The documented source code of a plug-and-play C++ class which (among other functionalities) calculates LDA directions is available upon request [1].

1 Pattern classification

1.1 General issue

The general matter is a pattern classification problem. It consists in classifying an object in a category (class). In the general case, the input data are :

- p classes of objects of the same type ;
- n observables defined for all the classes ;
- for each of the p classes, a sample of N_k objects, k being the class index.

There exists a more general case in which the observables are not necessarily defined for all the classes, and in which even the classes themselves may not be defined : there are pattern classification algorithms which are able to determine themselves the number of existing classes and their characteristics [2]. Having samples whose class is known is therefore not mandatory – but it is always better so as to obtain a good performance.

So the aim is the creation of an algorithm which, from the input data listed above, is able to classify a new object into one of the classes defined. In general, this is realized through 5 distinct phases :

- data collection ;
- classes characterization ;
- choice of a pattern classification algorithm ;
- training (or *learning*) ;
- tests.

After the test phase, any of the 4 previous phases can of course be changed according to the results obtained.

Figure 1 describes how a pattern classification algorithm works.

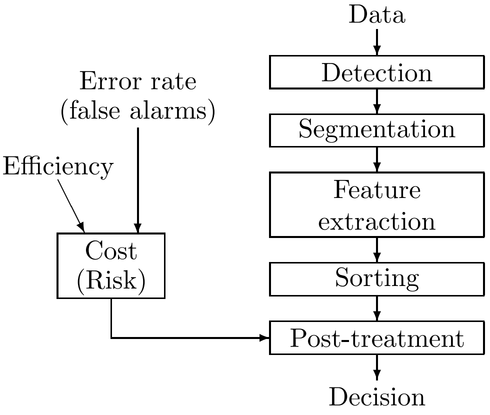


FIG. 1 – *Pattern classification algorithm*

The phase involving the detectors is the data collection, which is low-level information. In our case, it is the collection of the TPC points or of the hits in the Silicon detectors, for example.

The phases of segmentation and feature extraction transform the low-level information into mid-level information, the latter being generally smaller in size and more informative. In our case, segmentation corresponds for example to track and vertices reconstruction, and feature extraction is the calculation of the various cut variables, like the geometrical and kinematic parameters of the decay vertices. The segmentation phase is often the most difficult part to set up.

Classification, better called "sorting", is not the last phase. It consists in calculating high-level information from the previously mentioned mid-level information, most of the time only a handful of variables, not to say just one, but which are very informative. At this stage, the sorting is done, as two objects can be compared together.

Yet, the final decision can be taken only after the post-treatment phase, which takes into account an efficiency and a false alarms rate in the calculation of the decision. This decision corresponds to the minimization of a cost.

In our case, the number p of classes is 2, and from now they will be called *signal* and *background* (or *noise*). The signal is made of the real searched particles (Ξ or Ω in our case), while the background is made of all the other candidates (for the multistrange analysis : all the other xiVertex : combinatorial, correlations, other real particles,...).

[1] Current e-mail address : `julien.faivre@pd.infn.it`.

[2] This mode of operation is called unsupervised learning, as opposed to supervised learning for which the class to which each candidate of the training sample belongs to is known.

1.2 Problem of overtraining

The test phase is essential to obtain a performant algorithm, as its performance is not the same if calculated on the training sample or on a test sample. The latter is always worse than the former.

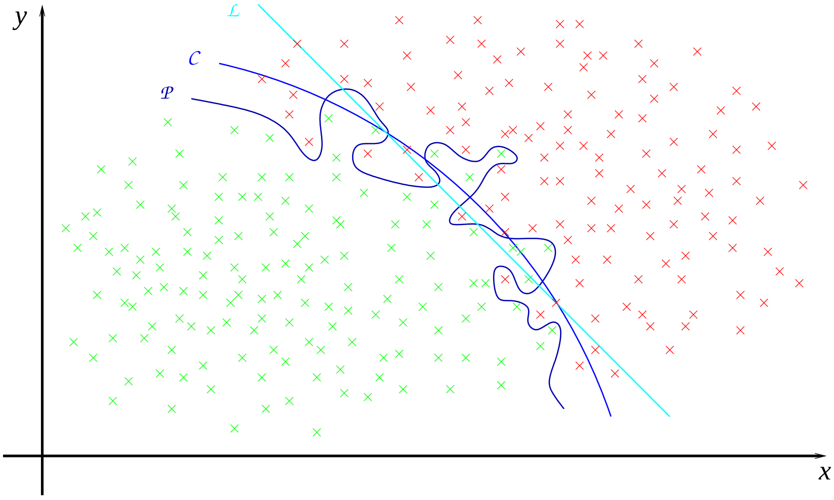


FIG. 2 – *Performance of various boundaries on their* training *sample*

This is illustrated by figure 2, in which the distributions of two classes are shown (in green and red) for the *training* sample, as well as three examples of border : a line $\mathcal{L}$, a simple curve $\mathcal{C}$ which describes a bit better the boundary between both classes, and a complex parametrization $\mathcal{P}$ that describes the samples almost candidate-by-candidate.

The result of those boundaries on a test sample will be very different : the line will have a fair performance and the simple curve will have a good one, but the performance of the complex curve will be bad.

The reason why is that the distribution of two samples is globally close to identical, but is locally different, because of their finite statistics and of possible systematic differences ; a pattern classification algorithm can therefore not be based on a too local description of the training samples. These observations are important in our case, as will be shown in part 5 (p. 12).

1.3 Estimation of the performance

No discriminancy criterion will be defined here, as we have a cost function at disposal, which will be used directly to tune the cuts.

In all that follows, S will refer to an amount of signal and N to an amount of background (noise), except when explicitly mentioned.

Let's first define the couple of variables that will be used in what follows as indicators of cuts' performance :

- the *amount of signal* S : it is the simplest indicator ;
- the *background rejection* $1-\varepsilon_N = \frac{N_{\text{removed by cuts}}}{N_{\text{pre-cuts}}}$ is the proportion of background that is rejected by the cuts ;
- the *efficiency* or *sensitivity* or *detection probability* $\varepsilon_S = \frac{S_{\text{post-cuts}}}{S_{\text{pre-cuts}}}$ is the proportion of signal that is kept by the cuts ;
- the *purity* or *specificity* $\frac{S}{S+N}$ is the proportion of kept candidates that actually *are* signal ;
- the *false alarms rate* $\frac{N}{S+N}$ is the proportion of kept candidates which are actually background ;
- the *signal to noise ratio* $\frac{S}{N}$;
- the *relative uncertainty*, connected to the inverse of the significance.

All these variables are 2 by 2 independent (except for the purity, the false alarms rate and the signal to noise ratio), and hence bring varied information.

The expression of the relative uncertainty changes with respect to the analysis. Its most general expression is $\frac{\sigma_S}{S}$ in the framework of signal counting. Details about the expression of σ_S in our case can be found in § V-2.4 and V-3.2 of [4].

The cost function used will be the relative uncertainty, as it is the indicator that directly guarantees the smallest possible statistical error on the result. It is yet common to show other indicators as well to determine the performance of a method. Doing so requires the association of two of them, such as :

– signal with respect to the signal to noise ratio ;
– signal with respect to purity : this diagram is strictly equivalent to an efficiency-purity diagram, and also strictly equivalent to the diagram mentioned below ;
– efficiency with respect to the false alarms rate : this diagram is called "ROC curve" (Receiver Operating Characteristic), and is widely used when it comes to comparing different pattern classification methods ;
– relative uncertainty with respect to signal.

Changing the cuts obtained by a given method defines in such diagrams a zone made of the points that are reachable by this method. This zone may be a surface (case of the classical cuts) or a curve (case of LDA). In a signal-S/N or an efficiency-purity diagram, a movement along the curve (or along the border of the surface) inducing an improvement of one of the variables results in a deterioration of the other one. The relative uncertainty being the cost function, the behaviour is different in a diagram showing this latter variable versus the signal : in such a diagram, the curve is a decreasing, then increasing function [1] which has a global minimum. The latter corresponds to the searched optimal cut.

2 Observables : cuts variables

The variables used as characteristics of a class may be chosen amongst the parameters which are directly accessible, or may be made from those ones. In the most general case, a variable used in a pattern classification method may be the discriminating output variable of a previous and possibly different pattern classification method. The number of variables to use is a study by itself. It should in general be as high as possible, so as to have the highest possible discriminancy, but it may be limited for statistics or processing time reasons. Methods exist to reduce this number of variables while avoiding a drop in discriminancy (see the end of section 5).

In this section, we give an example of the observables that have been chosen for a Ξ and Ω analysis by topological reconstruction in `Au-Au` 200 GeV collisions.

We have employed 25 variables, almost all directly accessible, which can be split into four categories : 10 geometrical variables, 11 pointing angles, one kinematic variable, and the 3 numbers of hits in the TPC. Because of the intrinsic characteristics of the linear discriminant analysis method, even multicut (*cf.* § 5), the distributions (at minimum the signal's) have to show only one peak whenever possible, for the method to be fully efficient. It is also preferable to use reasonably well shaped distributions, e.g. the pointing angle value is better than using its cosine, as the cosine function will flatten everything towards 1, which may cause the algorithm to fail using that variable in the optimization, since the peak would be extremely narrow.

2.1 Usual geometrical cuts

The 2-dimension geometry of a v0Vertex is shown in figure 3. The charged tracks are curved by the axial magnetic field (here perpendicular to the figure plane), and the reconstruction is imperfect because of the finite resolution of the detectors : the tracks of the two decay daughters don't cross and the trajectory of the reconstructed V0 doesn't meet the primary vertex.

In this figure, the trajectory of each of the daughter particles is a thick solid line, while the extrapolations towards the primary vertex are thin solid lines. The trajectory of the reconstructed V0 is a thick dashed line.

[1] The optimum being taken as a reference, tightening or loosening the cuts makes the amount of signal drop more rapidly or raise less rapidly than the error bar, in the first case because of the small amount of background, and in the second case because the amount of background raises faster than that of signal.

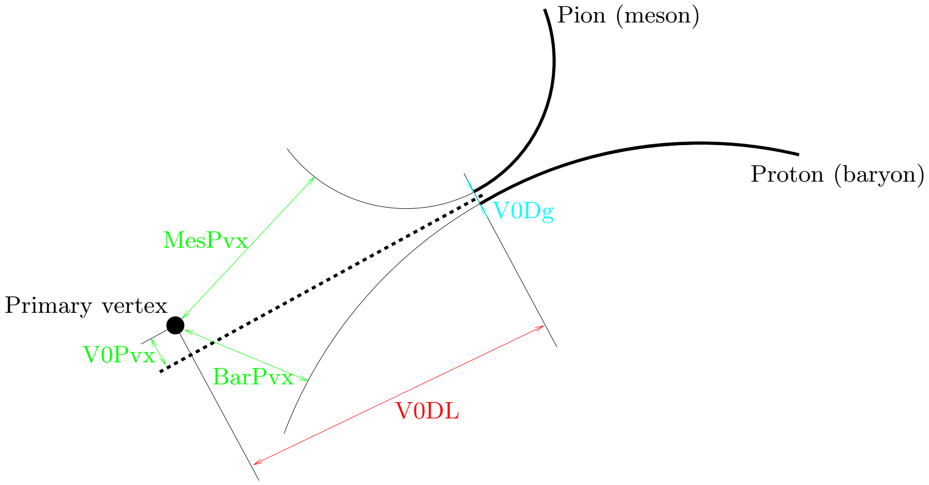


FIG. 3 – *2-dimension geometry of a v0Vertex ; see text for the notations*

Five characteristic distances constitute geometrical cuts which can be used to discriminate the background (fortuitous associations of tracks) and the signal (Λ or K_s^0 for example). These characteristic variables can generally be divided into 3 groups :

- the distances of closest approach between the daughters, in blue ;
- the distances of closest approach between a particle and the primary vertex, in green ;
- the distances between the vertices (decay vertices and primary vertex), including the decay lengths, in red.

The abbreviation "DCA" will sometimes be used instead of "distance of closest approach".

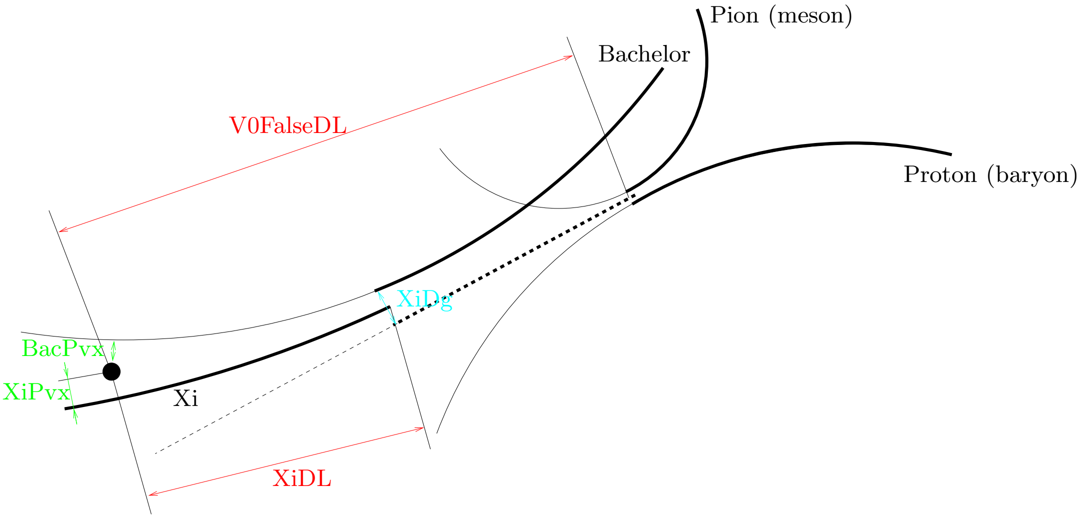


FIG. 4 – *2-dimension geometry of a xiVertex ; see text for the notations*

This splitting into three groups can also be made for the xiVertex, for which five additional variables are defined, which makes a total of 10. They are shown in figure 4.

The 10 geometrical cuts used are :

- the distance of closest approach `V0Dg` between the Λ daughters ;
- the distance of closest approach `XiDg` between the Ξ or Ω daughters ;

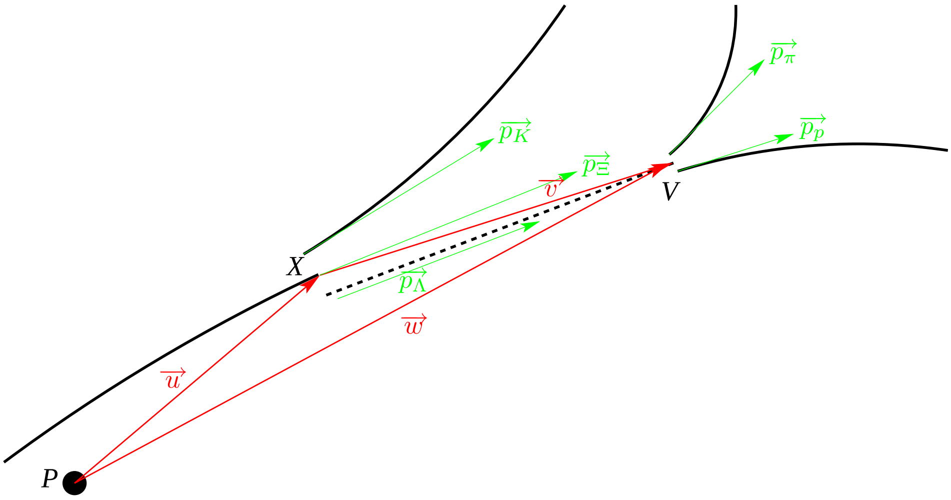


FIG. 5 – *Vectors used in the definition of the pointing angles*

- the Ξ or Ω decay length `XiDL` ;
- the Λ decay length `V0TrueDL` ;
- the false Λ decay length `V0FalseDL`, i.e. the distance between its decay vertex and the primary vertex ;
- the distance of closest approach `XiPvx` between the Ξ or Ω and the primary vertex ;
- the distance of closest approach `V0Pvx` between the Λ and the primary vertex ;
- the distance of closest approach `BacPvx` between the bachelor and the primary vertex ;
- the distance of closest approach `MesPvx` between the meson and the primary vertex ;
- the distance of closest approach `BarPvx` between the baryon and the primary vertex.

Most of these variables are partly correlated. E.g. the false Λ decay length is highly correlated with the Λ decay length and with the Ξ or Ω decay length. Or also the bachelor-to-primary distance of closest approach is correlated with the Ξ or Ω decay length.

2.2 Pointing angles

The pointing angles are angles which can be defined between the momentum of a particle and the direction given by two vertices. Let P, X and V be respectively the primary vertex, the Ξ or Ω decay vertex, and that of the Λ, and let's define those three vectors : $\overrightarrow{u} = \overrightarrow{PX}$, $\overrightarrow{v} = \overrightarrow{XV}$ and $\overrightarrow{w} = \overrightarrow{PV} = \overrightarrow{u} + \overrightarrow{v}$. The eleven pointing angles that are used are :

- $(\overrightarrow{u},\overrightarrow{p^X_\Xi})$, $(\overrightarrow{u},\overrightarrow{p_\Lambda})$, $(\overrightarrow{u},\overrightarrow{p^X_{Bac}})$, $(\overrightarrow{u},\overrightarrow{p^V_{Mes}})$, $(\overrightarrow{u},\overrightarrow{p^V_{Bar}})$;
- $(\overrightarrow{v},\overrightarrow{p_\Lambda})$, $(\overrightarrow{v},\overrightarrow{p^V_{Mes}})$, $(\overrightarrow{v},\overrightarrow{p^V_{Bar}})$;
- $(\overrightarrow{w},\overrightarrow{p_\Lambda})$, $(\overrightarrow{w},\overrightarrow{p^V_{Mes}})$, $(\overrightarrow{w},\overrightarrow{p^V_{Bar}})$.

The momenta of the Xi and of the bachelor are taken at the Xi decay vertex, those of the meson and of the baryon are taken at the V0 decay point. The various vectors are shown in figure 5, as an example of an Ω decay.

2.3 Cosine of the decay angle

The cosine of the decay angle, or $\cos\theta^*$, is usually used to discriminate the signal and the background or the correlations. It is defined as follows : let $\overrightarrow{p}$ be the momentum of the decaying particle, and $\overrightarrow{p_1}$ and $\overrightarrow{p_2}$ those of the daughter particles in the lab frame. Let $\overrightarrow{p_1^*}$ and $\overrightarrow{p_2^*} = -\overrightarrow{p_1^*}$ be these momenta in the center of mass rest frame. The cosine of the decay angle is defined as :

$$\cos\theta^* = \cos(\overrightarrow{p},\overrightarrow{p_1^*}) \tag{1}$$

The distribution of this variable shows strong peaks at -1 and $+1$ for background and correlations. It can for example be used to get rid of the Ξ when Ω is searched, or of the Λ when K_s^0 is searched. The $\cos\theta^*$ of the v0Vertex can also be used.

2.4 Number of hits in the TPC

The three last variables are the number of hits left in the TPC by each of the three tracks of the xiVertex. The characteristics of the background are indeed different for low or high numbers of hits in the TPC.

3 Learning samples

The method developed here is a supervised learning method, it therefore needs a sample of each class separately.

The background sample is made of candidates coming from the real data. They are mainly background, but the proportion of signal may be a nuisance, hence the latter has to be removed : this can be done by a simple invariant mass cut around the particle mass. Another, larger, invariant mass window allows to select only the candidates that are reasonably close to the particle mass, as it is in this area that the background has to be lowered.

The signal sample is obtained from the simulation : it is made of the embedding's associated candidates.

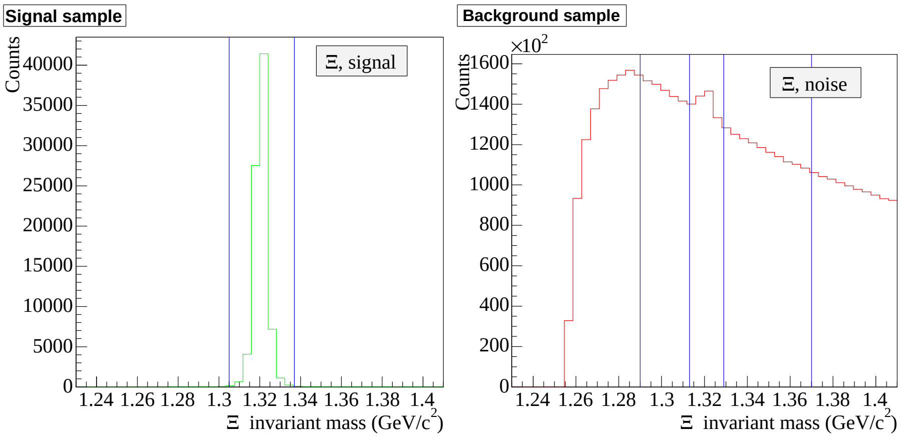


FIG. 6 – *$\Xi+\overline{\Xi}$ invariant mass distributions of the training samples : signal (left) and background (right) ; the limits of the various cut windows are shown in blue*

Figure 6 shows as an example the invariant mass distributions of the Ξ signal and background candidates and the cut windows used. The Ξ signal peak – excluded by the most internal window – is clearly visible in the background sample, although the cuts at this stage are still loose.

The training samples are dE/dx-filtered and the cuts that are not used in LDA are also applied, for LDA to have a better performance. Those cuts include : a minimal transverse momentum, a maximal rapidity, a maximal deviation of the reconstructed Λ mass compared to the PDG value, and also a loose filtering on the number of hits in the TPC for each track, so as to get rid of the worst tracks before running LDA training and filtering.

4 Principles of basic LDA

LDA stands for Linear Discriminant Analysis, and refers to a set of pattern classification methods which common property is to consider that each class fills, in the space defined by the n observables, a convex volume which boundaries are defined by hyperplanes.

4.1 Advantages of using LDA

4.1.1 Improvement in statistics brought with respect to the classical cuts

The principle of the LDA method is illustrated by the three drawings of figure 7. It has been supposed that 2 observables, x and y, were accessible to the observer, and the signal and background distributions have respectively been drawn in green and in red. The cyan blue zones are removed by the cuts, which are represented by the blue lines.

The two first drawings show the behavior of the classical cuts, i.e. steep cuts on one or several of the observables, and it has to be kept in mind that the number of background candidates is way higher than that of signal candidates. On the top plot, the cuts chosen are loose for the efficiency to be high, but, as a consequence, the pollution of the signal by the background is high. The middle plot shows tighter cuts which avoid having an overwhelming background, but the price to pay is a small efficiency.

LDA consists in cutting along a linear combination of all the observables, rather than along each of the observables. This linear combination is defined by an LDA direction (or axis). The result, shown in the bottom plot, is a better discrimination between both classes (signal and background). This translates into a more interesting position of the cuts in the efficiency-purity diagram than all the positions accessible to the classical cuts.

The algorithm consists in calculating the direction of this axis so as to have an optimal discrimination between the classes according to a given criterion. A hyperplane perpendicular to the axis is then associated to this maximal discriminancy and is called *best discriminancy hyperplane*. Three examples of criterion will be given in the next paragraphs, as well as the corresponding algorithms.

4.1.2 Easiness of cut tuning

A second interest of LDA is that the cuts can very easily be tuned to reach the optimal point.

The cuts optimization is realized by minimizing the relative error on the final result. It is therefore a matter of minimization of a function that is defined from the cuts space to $\mathbb{R}$.

In the case of the classical cuts, the dimension of the cuts space is the number of variables used. This space is therefore $\mathbb{R}^n$. But minimizing a function defined from $\mathbb{R}^n$ to $\mathbb{R}$ is very complex – and it is actually empirically realized.

From the fact that it works with a linear combination of the observables, LDA brings a transformation from the $\mathbb{R}^n$ space to a curve that is equivalent to $\mathbb{R}$, which means that one now has to deal with the easy minimization of a function from $\mathbb{R}$ to $\mathbb{R}$. Indeed, one only needs to calculate the relative error as a function of the LDA cut tightening or loosening and to determine its minimum, given that, on top of this, this function decreases and then increases, and therefore doesn't lead to any ambiguity on the position of the minimum.

4.1.3 Cuts for particular conditions

Some areas of the phase space have a different proportion of background and hence may need tighter or looser cuts than the "usual" cuts.

In the case of e.g. a cut loosening, the classical cuts require to find again the minimum of a function defined from $\mathbb{R}^n$ to $\mathbb{R}$, while with LDA, a simple loosening of the LDA cut, until the new minimum is reached, makes it. The amount of time that is saved is considerable, as this operation needs to be done for each new collision system or collision energy, each centrality range, and, if wished, for low and high transverse momenta, where statistics is poor.

But it is also easy to calculate a new set of cuts, optimized for a specific area of the phase space, by building the training samples with candidates which belong only to this area, typically low- or high-$p_\perp$.

4.2 Fisher criterion

The optimal direction found obviously depends on the criterion used for its calculation. The most frequently used is the Fisher criterion, which gives what is called Fisher LDA, introduced by Ronald Fisher in 1936 [5] [1].

[1] It took me a while to find the electronic version of the original paper so I share the URL : you can get it on the Adelaide library website : `www.library.adelaide.edu.au/digitised/fisher/stat_math.html`

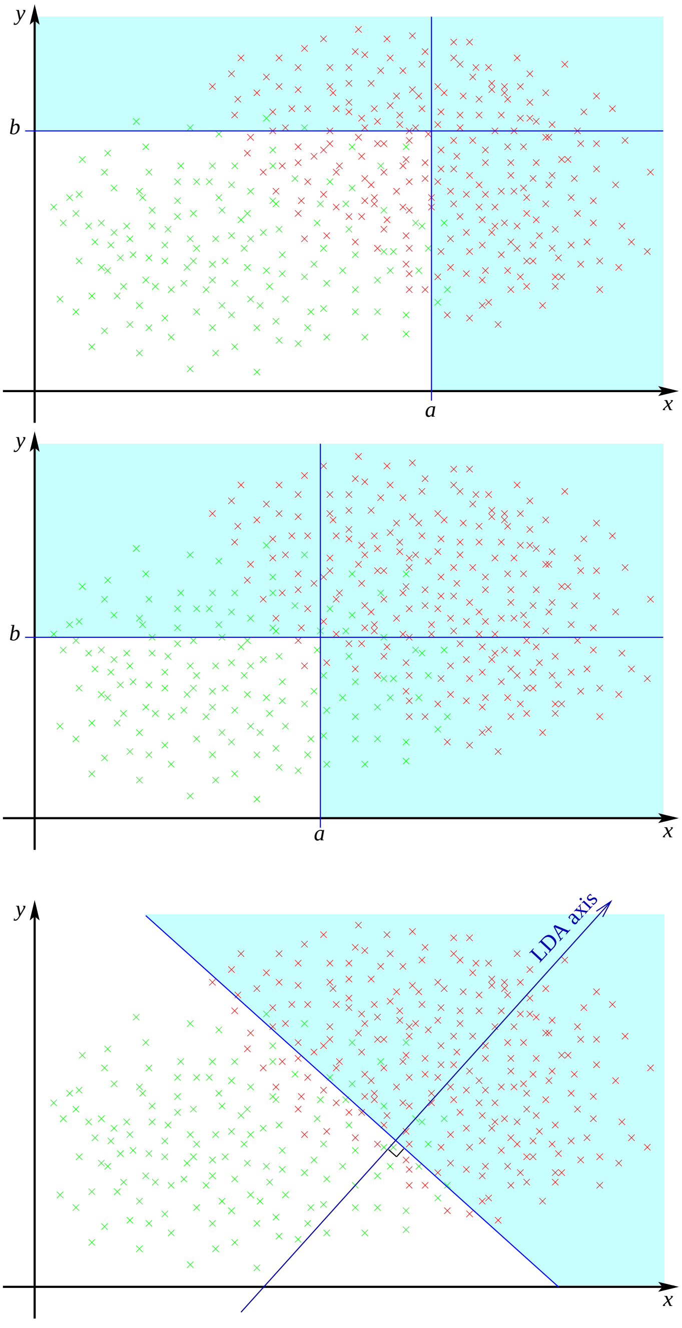
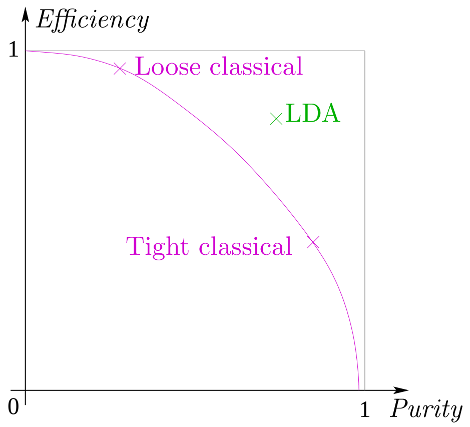


Fig. 7 – *Principle of basic* LDA *: example with 2 variables.*

When classical cuts are applied (top and middle drawings), one has to choose between a good efficiency and a good signal to noise ratio.
An LDA *cut (bottom drawing) gives a better compromise.*

On the efficiency-purity graph below, the zone of the best possible points reachable by tuning the classical cuts is drawn in mauve, while the position of a point obtained by LDA *cuts is in green.*



The advantage of the Fisher criterion is that, on top of being easy to settle, it gives the exact expression of the direction of the LDA vector, without a need for an optimization algorithm. There is indeed a maximization, but the solution is analytical.

Let's call Δ a line and $\overrightarrow{u}$ its driving vector, and let's project the points of the learning samples on it. Let μ_1 and μ_2 be the means of the distributions of the projected points for classes 1 and 2 respectively, and σ_1^2 and σ_2^2 be the "dispersions" (variances not normalized by the number of observations) : $\sigma_k^2 = \sum_{\overrightarrow{x} \in \mathcal{D}_k} (\overrightarrow{u}.\overrightarrow{x} - \mu_k)^2$. The Fisher criterion consists in requiring that the means of the distributions be as far as possible one from the other and that their widths be as small as possible, for the overlap between the distributions to be minimal. This translates into a maximization of

$$\boxed{\lambda(\Delta) = \frac{|\mu_1(\Delta) - \mu_2(\Delta)|^2}{\sigma_1^2(\Delta) + \sigma_2^2(\Delta)}} \qquad (2)$$

and has some analogy with the resolution power of a telescope, as shown by figure 8.

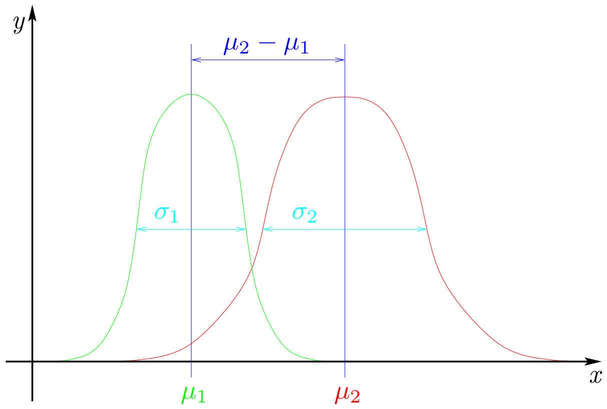


Fig. 8 – *Analogy between the Fisher criterion and the resolution power of a telescope*

On this figure, the variable x is respectively the LDA coordinate (i.e. the curvilinear abscissa on the line Δ, obtained from the scalar product) or the position (spatial or angular) depending whether we consider LDA or a telescope ; y stands for a count number or for a received amount of light. A good separation power requires not only that the means be far, but also that the Airy stains don't overlap, i.e. that the width of the distributions be small with respect to the distance between the means.

The LDA direction is therefore :

$$\Delta_{LDA} \quad = \quad \Delta \quad / \quad \lambda(\Delta) = \lambda_{max}$$

In the following, we will call n the number of observables, $\overrightarrow{u}$ the normalized n-coordinate vector which drives the line Δ (which itself characterizes, together with the cut value, the hyperplane which plays the role of a border between both classes), and N_k, $k \in \{1; 2\}$ the number of objects in the training sample of class k. The sets of the sample candidates will be called $\mathcal{D}_k$.

Let $\overrightarrow{x}$ be an observation (so an n-coordinate vector) : its projection on line Δ is simply the scalar product with $\overrightarrow{u}$ and writes : $\overrightarrow{x}.\overrightarrow{u}$. The mean of a distribution is :

$$\overrightarrow{m_k} = \frac{1}{N_k} \sum_{\overrightarrow{x} \in \mathcal{D}_k} \overrightarrow{x}$$

and the mean of the projection on $\overrightarrow{u}$ is therefore :

$$\mu_k = \frac{1}{N_k} \sum_{\overrightarrow{x} \in \mathcal{D}_k} \overrightarrow{x}.\overrightarrow{u} = \overrightarrow{m_k}.\overrightarrow{u}$$

The distance between the projected means can now be calculated :

$$|\mu_1 - \mu_2| = |(\overrightarrow{m_1} - \overrightarrow{m_2}).\overrightarrow{u}|$$

Using the transposed matrices – we write here ${}^tM$ for the transposed matrix of M –, this latter formula can be re-written into $|{}^t\overrightarrow{u}(\overrightarrow{m_1} - \overrightarrow{m_2})|$. Hence we obtain :

$$\begin{aligned}(\mu_1 - \mu_2)^2 &= \left({}^t\overrightarrow{u}(\overrightarrow{m_1} - \overrightarrow{m_2})\right)^2 \\ &= \left({}^t\overrightarrow{u}(\overrightarrow{m_1} - \overrightarrow{m_2})\right)\left({}^t(\overrightarrow{m_1} - \overrightarrow{m_2})\overrightarrow{u}\right) \\ &= {}^t\overrightarrow{u}(\overrightarrow{m_1} - \overrightarrow{m_2}).{}^t(\overrightarrow{m_1} - \overrightarrow{m_2})\overrightarrow{u} \\ &= {}^t\overrightarrow{u}S_B\overrightarrow{u}\end{aligned}$$

with $S_B = (\overrightarrow{m_1} - \overrightarrow{m_2}).{}^t(\overrightarrow{m_1} - \overrightarrow{m_2})$ the between-class scatter matrix.

A matrix S_W can similarly be defined to calculate $\sigma_1^2 + \sigma_2^2$. S_W is actually the within-class scatter matrix. The contributions of the various classes being dissociable, let's calculate σ_k :

$$\begin{aligned}
\sigma_k^2 &= \sum_{\overrightarrow{x} \in \mathcal{D}_k} (\overrightarrow{u}.\overrightarrow{x} - \mu_k)^2 \\
&= \sum_{\overrightarrow{x} \in \mathcal{D}_k} \left({}^t\overrightarrow{u}(\overrightarrow{x} - \overrightarrow{m_k})\right)^2 \\
&= \sum_{\overrightarrow{x} \in \mathcal{D}_k} {}^t\overrightarrow{u}(\overrightarrow{x} - \overrightarrow{m_k}).{}^t(\overrightarrow{x} - \overrightarrow{m_k})\,\overrightarrow{u} \\
&= {}^t\overrightarrow{u}\,S_k\,\overrightarrow{u}
\end{aligned}$$

with $S_k = \sum\limits_{\overrightarrow{x} \in \mathcal{D}_k} (\overrightarrow{x} - \overrightarrow{m_k}).{}^t(\overrightarrow{x} - \overrightarrow{m_k})$, and we have very simply $S_W = S_1 + S_2$.

Thus we can now write the Fisher criterion matricially :

$$\lambda(\Delta) = \lambda(\overrightarrow{u}) = \frac{|\mu_1 - \mu_2|^2}{\sigma_1^2 + \sigma_2^2} = \frac{{}^t\overrightarrow{u}\,S_B\,\overrightarrow{u}}{{}^t\overrightarrow{u}\,S_W\,\overrightarrow{u}} \tag{3}$$

This expression being invariant by transformation of $\overrightarrow{u}$ into $\alpha\overrightarrow{u}, \alpha \in \mathbb{R}$, its maximization is equivalent to the maximization of ${}^t\overrightarrow{u}S_B\overrightarrow{u}$ under the condition ${}^t\overrightarrow{u}S_W\overrightarrow{u} = 1$. The Lagrange multiplier (here, ω) method can then be used : the maximum is reached when :

$$\forall j \in < 1; n > \quad \frac{\partial}{\partial u_j}\left({}^t\overrightarrow{u}S_B\overrightarrow{u} - \omega({}^t\overrightarrow{u}S_W\overrightarrow{u} - 1)\right) = 0 \tag{4}$$

The development of the matrix products gives, for $S = S_W$ or $S = S_B$:

$$\begin{aligned}
\frac{\partial\,{}^t\overrightarrow{u}S\overrightarrow{u}}{\partial u_j} &= \frac{\partial}{\partial u_j}\left(\sum_{l=1}^{n}\sum_{k=1}^{n} s_{l,k}u_l u_k\right) \\
&= \sum_{l=1}^{n}\sum_{k=1}^{n} s_{l,k}\left(\delta_{j,l}u_k + \delta_{j,k}u_l\right) && \text{(Swapping the sums and the derivation)} \\
&= 2\sum_{k=1}^{n} s_{j,k}u_k && \text{(Because } S_W \text{ and } S_B \text{ are symmetric)}
\end{aligned}$$

Equation (4) can then be re-written :

$$\begin{aligned}
(4) \quad &\Leftrightarrow \quad 2S_B u - 2\omega S_W u = 0 \\
&\Leftrightarrow \quad S_W^{-1}S_B u = \omega u
\end{aligned}$$

We therefore proved that a vector $\overrightarrow{u}$ maximizing expression (3) obeys :

$$\exists\,\omega \in \mathbb{R} \quad / \quad S_W^{-1}S_B\overrightarrow{u} = \omega\overrightarrow{u}$$

${}^t(\overrightarrow{m_1} - \overrightarrow{m_2}).\overrightarrow{u}$ being a scalar, $S_B\overrightarrow{u}$ is always collinear to $\overrightarrow{m_1} - \overrightarrow{m_2}$, and the expression becomes :

$$\exists\,\xi \in \mathbb{R} \quad / \quad S_W^{-1}(\overrightarrow{m_1} - \overrightarrow{m_2}) = \xi\overrightarrow{u}$$

A normalization of $S_W^{-1}(\overrightarrow{m_1} - \overrightarrow{m_2})$ gives the driving vector of the LDA axis :

$$\boxed{\overrightarrow{u} = \frac{S_W^{-1}(\overrightarrow{m_1} - \overrightarrow{m_2})}{\|S_W^{-1}(\overrightarrow{m_1} - \overrightarrow{m_2})\|}} \tag{5}$$

The Fisher criterion can hence be analytically resolved, which thus provides a low calculation time method, and avoids the need for the implementation of a numerical optimization algorithm.

As it has been said in paragraphs 1.3 (p. 3) and 4.1 (p. 8), determining the value of the cut along the axis is done by adjusting this cut so as to obtain the lowest relative uncertainty. There is only one cut to change, so the process is simple and fast.

4.3 Problems which show up

Using the Fisher criterion, even though it is satisfactory for most of the applications, raises several problems in our case.

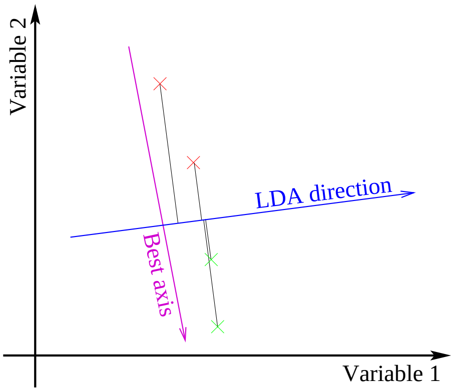


Fig. 9 – *Case of a denominator tending towards zero in the Fisher criterion*

The first point to mention is that the LDA axis that is determined with the Fisher criterion gives the best discriminancy hyperplane only when the distributions of both classes are gaussian [2] : this criterion is adapted to the “simple” distributions which are almost completely described by their mean and their standard deviation. This implies that for the other distributions – and particularly in our case –, it is possible to find a criterion which gives a better discriminancy than Fisher. From the arguments stated above comes the need for taking care of the local variations of the distributions. The Fisher criterion indeed takes into account the distributions only globally, as this is the only information carried by their mean and standard deviation. A better discriminancy therefore requires a local description of the distributions, yet without falling into the excess represented by the curve $\mathcal{P}$ in figure 2 p. 3.

Then, we want to apply LDA not to simply separate two classes, but to actually *extract* candidates of a class (the signal) out of those of another class (the background) which has no interest for us, which statistics is overwhelming (orders of magnitude higher than that of the signal population [1]), and which, contrary to the signal, populates almost the whole cut space. The background population which lies above the signal is therefore only a local part of the whole background distribution, and can not be correctly described by the usual global parameters (mean and standard deviation). The two main differences with the usual applications are this difference in statistics, and the fact that the range of the background distribution covers that of the signal distribution. In other words, some background candidates may have a geometry that no signal candidate can have, but all signal candidates have a geometry that a background candidate can have.

And finally, the Fisher criterion consisting into maximizing a ratio, it sometimes happens that the denominator reaches values that are close to zero (this is the case of distributions which standard deviation along some direction is small, for example when there is a strong linear correlation between two variables). Figure 9 shows a real case, for which the blue axis – the best direction Fisher-wise – is almost perpendicular to the direction that one would wish to find, materialized by the mauve axis. Yet, this problem can be solved by a whitening of the data (removal of the linear correlations), which principle will be explained in § 5.5.4.

5 LDA improvements

5.1 Multicut-LDA

The second problem in the list of the previous subsection, that is to say the high predominance of the background over the signal, can be solved by the multicut-LDA. This method also allows, to a certain extent, a better management of the first problem, i.e. taking care of the local parts of the distributions, although a real and much better solution will be brought in the next paragraph by changing the criterion.

Multicut-LDA consists in applying successively several LDA cuts. The first cut is determined by a learning over all the candidates of both samples. A cut value is then determined according to criteria which will be described

[1] After the reconstruction cuts, there is about 25 000 times more background than signal in a central event for the Ξ ; this factor reaches about 400 000 for the Ω.

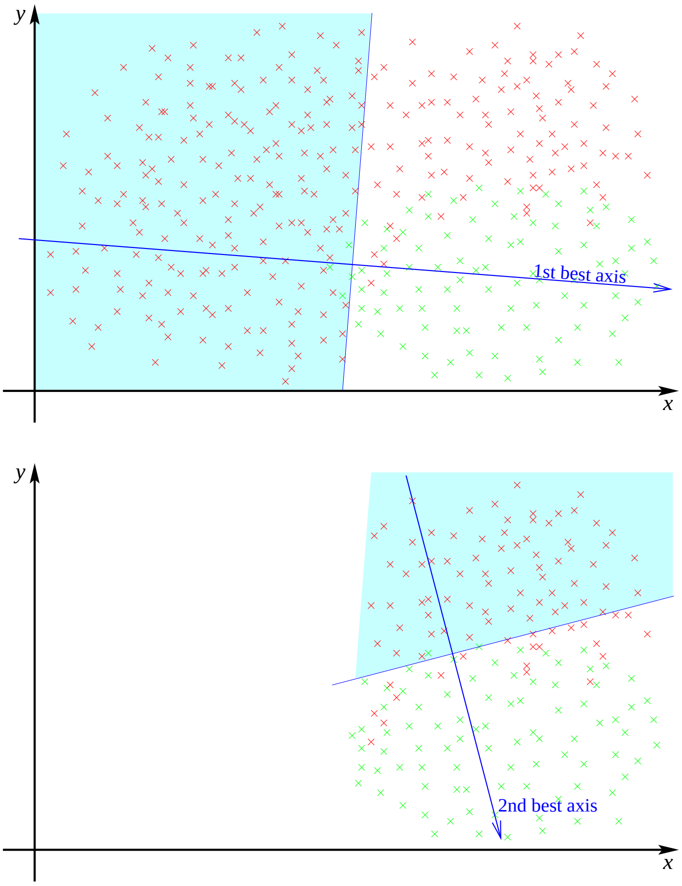


Fig. 10 – *Mechanism of the multicut-*LDA *method :*

The first step is illustrated by the top figure : a first LDA *direction (in blue) is calculated with all the candidates of the training samples. The cut value is determined in a way that its efficiency on the signal is close to 100 %. The area cut is shown in cyan-blue.*

The next step is described by the bottom figure : the candidates of the former cyan zone have been removed from the training samples, and a second LDA *direction is calculated with the remaining candidates. The cut value is determined as for the previous one. A third cut can then be calculated, and so on.*

in paragraph 5.4 (p. 17). This first cut is applied to the learning samples, and a second LDA direction is calculated with the remaining candidates. A cut value for this second direction is then calculated, and so forth. Figure 10 illustrates the mechanism of this method.

Multicut-LDA therefore provides a set of LDA directions, each one being a vector $\overrightarrow{u_i}$ of the observables space (n coordinates). It also provides a cut value c_i associated to direction $\overrightarrow{u_i}$. The value of c_i depends on $\overrightarrow{u_i}$, and the direction $\overrightarrow{u_i}$ is a function of $\overrightarrow{u_{i-1}}$ and c_{i-1}. Each pair (direction , cut) defines a hyperplane, and this set of hyperplanes demarcates a connex and even convex shape, by construction [1], in which the candidates are considered as being signal.

The number of LDA cuts to apply can be determined only empirically. There is obviously no lower limit to the number of cuts, but the higher this number, the more the extraction of the signal out of the background will be possible. This technique enables to get rid of the purely linear character of the LDA boundary between the background and the signal, yet without loosing the linearity of the algorithm itself.

There are however two upper limits. The first one comes from the fact that a too local description of the distributions leads to a poor discriminancy, as explained in paragraph 1.2 (p. 3) (see in particular figure 2). The second one is due to the fact that as one goes along applying the LDA cuts, the number of candidates in the training samples decreases, and may at some step become insufficient to calculate correctly the next LDA direction.

Paragraph 5.4 (p. 17) explains how the first of these two upper limits can be determined, *via* the minimum number of candidates removed by each cut, which directly determines the second upper limit. It is the latter which, for the Ω, will limit the number of LDA cuts used. For the Ξ, this number is limited by the first condition.

[1] The method can easily be modified for the convex zone to become the area which is removed rather than selected, but the region where the signal is more likely to be is more probably convex than concave in the general case. The change to make is a replacement of the cut criterion “Keep x if $(x > c_1).(x > c_2).\cdots$” by “Keep x if $(x > c_1) + (x > c_2) + \cdots$”, with . and + the logical symbols and c_i the value of the i^{th} cut, associated with a change of the LDA criterion (optimized II).

5.2 Optimized criteria

The other problems created by Fisher LDA – global description of the distributions and small denominator "artificially" increasing the criterion to be maximized – can be solved by changing the criterion. This of course doesn't imply to reconsider multicut-LDA, as this method doesn't depend on the criterion that is used to determine the LDA directions themselves. Actually, multicut-LDA is even made more performant by replacing the Fisher criterion with a criterion that takes into account the local parts of the distributions.

Two criteria, which we will call *optimized*, can be defined. The name *optimized* comes from the fact that, by construction, they are exactly what we search : a criterion allowing the extraction of a small quantity of signal out of an overwhelming background, taking the local, and not global, behaviour of the distributions into account : such criteria are perfectly suited for the multicut-LDA, which thereby benefits of an optimal use of the candidates located in the region to be cut.

Here are these two criteria, formulated for the calculation of the direction of the i^{th} LDA vector :

- Optimized criterion I : given an efficiency of the i^{th} LDA cut on the signal, maximization of the amount of background removed ;
- Optimized criterion II : given an efficiency of the i^{th} LDA cut on the background, minimization of the amount of signal removed.

Their formulation is antisymmetric for the signal and the background, but we haven't tested if these two criteria are equivalent.

The reason why is that they require a sorting of the table containing the training sample candidates of the class on which the efficiency of the cut is known (actually imposed), and this at each step of the optimization. For the criterion I, the table that is sorted is that of the signal ; for the criterion II it is that of the background. But when the multicut-LDA method is used, the number of background candidates used is often way larger than that of the signal candidates, as, the efficiency of each cut being a lot smaller for the background than for the signal, the calculation of the LDA directions needs to be begun with a very large sample of background. As a consequence, searching the directions with criterion II is a lot longer, and a test has shown that the calculation time needed is completely prohibitive.

In the following, we will therefore use only the optimized criterion I. In a mean term future, it could be possible to create a program that uses the optimized criterion II, firstly by reading the data for each cut so as to keep the number of background candidates used in the calculation constant, and secondly by (considerably) reducing the number of background candidates used for the first cuts, the consequence of such a restriction being a possible worse determination of these directions, because fewer candidates will be used.

Let's finally mention that because of a too low statistics of the background sample, the limit on the number of LDA cuts mentioned in the previous subsection may be reached (in our study, it is the case for the Ω but not for the Ξ), i.e. there are not enough background candidates in the training sample to calculate other directions, while the signal to noise ratio isn't satisfactory yet. In such a case, it may be judicious to determine the last direction with the Fisher criterion instead of the optimized criterion ; firstly because at this stage, the amounts of signal and background should have reached about the same order of magnitude, and secondly because for the last cut – tighter than the previous ones –, a global, rather than local, criterion is necessary. The problem of the denominator tending towards 0 may still counterbalance the advantage of using the Fisher criterion though.

5.3 Algorithm of the optimized criterion I

5.3.1 Function to maximize

Contrarily to the Fisher criterion, using an optimized criterion requires the implementation of a maximization algorithm (or minimization, depending on the criterion chosen ; from now on we will consider the case of criterion I, but the algorithm is also valid for criterion II).

Let ε_{S_i} be the given efficiency of the i^{th} cut on the signal and $\mathcal{D}_{S_i}$ the set of candidates of the signal sample after the $i-1$ first cuts. If 1 is assigned to *true* and 0 to *false*, the number of signal candidates removed by cutting at value c_i along the axis $\overrightarrow{u_i}$ is :

$$S_i - S_{i+1} = (1-\varepsilon_{S_i})S_i = \sum_{\overrightarrow{x}\in\mathcal{D}_{S_i}} (\overrightarrow{x}.\overrightarrow{u_i} < c_i)$$

with S_i the number of signal candidates (in the training sample) used to determine the i^{th} direction. The value of c_i is therefore determined so as to obey the following equality :

$$1 - \frac{\displaystyle\sum_{\vec{x} \in \mathcal{D}_{S_i}} (\vec{x}.\vec{u_i} < c_i)}{S_i} = \varepsilon_{S_i} \tag{6}$$

So the table sorting – by value of $\vec{x}.\vec{u_i}$ – mentioned earlier happens here. If more statistics was needed or if the optimized criterion II needed to be tried, it would still be possible to fit within a reasonable amount of CPU time by using only a fraction of the signal (resp. background) sample to search the value of c_i, under the hypothesis that this fraction is representative of the whole population, which is likely to be the case.

If the efficiency ε_{S_i} was actually used as an input data, this would cause some shifting of its value. But using directly the numbers $S_i - S_{i+1}$ is more judicious, not only because it would clear out this minor problem, but also because it allows to control the "locality degree" of the optimized criterion. This "locality degree" is determined by : 1°) the proportion of S_i that is removed, 2°) the number of candidates that are removed. The first one needs to be higher than the typical size (in number of candidates) of the statistical fluctuations for a sample of size S_i, for the algorithm not to trig on one of those fluctuations, and the second one needs to be higher than some fixed absolute number which ensures that the candidates that are removed are numerous enough to be really representative of the actual shape of the signal distribution in the area that is cut. Examples of numerical values are given in § 5.4.

The function f that is maximized is of course the number of background candidates that are removed by the cut ; hence we can write

$$\begin{aligned} f \quad : \quad & \mathbb{R}^n \longrightarrow \mathbb{N} \\ & \vec{u_i} \longmapsto \sum_{\vec{x} \in \mathcal{D}_{B_i}} (\vec{x}.\vec{u_i} < c_i) \end{aligned} \tag{7}$$

where $\mathcal{D}_{B_i}$ is the set of candidates of the background sample after the $i-1$ first cuts.

The efficiency ε_{S_i} being fixed (it is a chosen parameter), the optimization consists in maximizing f as a function of $\vec{u_i}$, knowing that the value of c_i depends on $\vec{u_i}$ (so it needs to be recalculated at each step).

5.3.2 Maximization algorithm

The algorithm chosen to maximize f consists in varying each coordinate of the vector $\vec{u}$ at a time [2]. If the set of the possible vectors $\vec{u}$ is represented by an n dimension space of which a base is made of the normed vectors $\vec{x_j} = (\delta_{1,j},\delta_{2,j}, \ldots ,\delta_{n,j})$ along the directions of the n observables[1], finding the maximum of f on this space with this method is equivalent to moving step by step in this space along a vector collinear to one of the $\vec{x_j}$.

Such an algorithm is simple to set up, but its drawback is that it may converge to a local maximum instead of the searched global maximum[2]. This problem is partially resolved by the initial condition : the natural start vector for this algorithm is the direction found with the Fisher criterion. This guarantees that the final result will necessarily be better than with Fisher, and that the algorithm starts in a zone in which the global maximum has a reasonable probability to be.

Technically, the algorithm is made of several imbricated loops. Here is their list, from the most external one to the most internal :

- loop over the variation step size of $\vec{u}$ (smaller and smaller step size) ;
- "infinite" loop out of which the program exits when the vector $\vec{u}$ doesn't move anymore (the maximum has been found for the considered variation step size) ;
- loop over j (a coordinate $\vec{x_j}$ of $\vec{u}$ at a time is changed) ;
- "infinite" loop out of which the program exits when the vector doesn't move anymore (the maximum has been reached for a variation of the j^{th} coordinate only).

Let's name u_j, $j \in< 1,n >$ the n coordinates of $\vec{u}$. This latter loop – research of the maximum, for a given step and a given vector $\vec{x_j}$ – consists in modifying $\vec{u}$ by changing only[3] its coordinate u_j.

[1] δ stands here for the Kronecker symbol : $\delta_{i,j} = 1 \Leftrightarrow i = j$; $\delta_{i,j} = 0$ otherwise.

[2] This drawback is shared by most of the maximization algorithms. It may be avoided by using for example a genetic algorithm.

[3] Except that the vector is re-normalized afterwards.

The variation step is the angle between $\overrightarrow{u}$ and the modified vector $\overrightarrow{v}$ in the n dimension space : the variations are thus uniform. The variation of u_j is calculated as a function of this angle. Let δ_j be the variation of this coordinate u_j, that is to say :

$$\begin{cases} \overrightarrow{u} = (u_1,u_2,\cdots,u_j,\cdots,u_n) \\ \overrightarrow{v} = (u_1,u_2,\cdots,u_j+\delta_j,\cdots,u_n) \end{cases}$$

Since $\|\overrightarrow{u}\| = 1$, we have :

$$(\overrightarrow{u}.\overrightarrow{v})^2 = \|\overrightarrow{v}\|^2 \cos^2\alpha$$

Expressing the vectors as a function of the u_i and of δ_j and using $\sum_{i\neq j} u_i^2 = 1-u_j^2$, we come to the following formula :

$$\delta_j^2(u_j^2-\cos^2\alpha) + 2u_j\delta_j\sin^2\alpha + \sin^2\alpha = 0$$

The expression of δ_j as a function of the variation angle α is then :

$$\delta_j = \frac{-2u_j\sin^2\alpha \pm \sqrt{1-u_j^2}\,\sin(2\alpha)}{2(u_j^2-\cos^2\alpha)} \tag{8}$$

On top of the fact that a maximum is reached, this loop may be ended by another circumstance : as shown in figure 11, the maximum may never be reached by changing only one coordinate. In such a case, the loop is stopped when the angle between the vector to change and the driving vector of the axis of the coordinate that is being varied is smaller than the (angular) variation step of the vector.

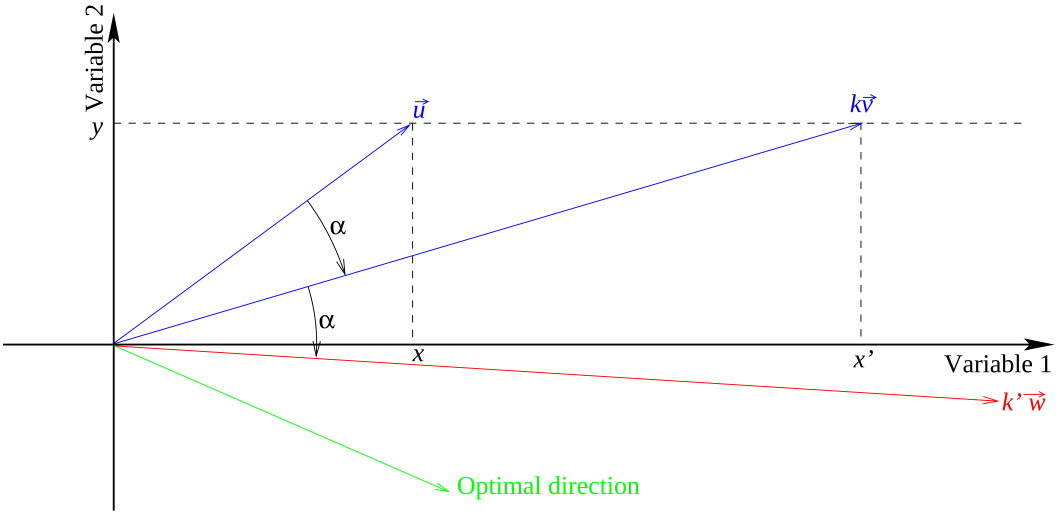


FIG. 11 – *Limit cone for the variation of the vector $\overrightarrow{u}$: the optimal direction is in green and the variation step of the LDA vector is the angle α. At some step of the optimization, the LDA vector is the unitary vector $\overrightarrow{u}$ drawn in blue. At the next step, it is vector $k\overrightarrow{v}$ ($\overrightarrow{v}$ unitary vector), of which only the first coordinate x has been changed with respect to $\overrightarrow{u}$. The next step, going from $\overrightarrow{v}$ to a vector that is collinear to the red vector $\overrightarrow{w}$, is impossible to pass if only the first coordinate is changed. It is therefore necessary to stop varying this coordinate and move on to the next one.*

The variation step is taken equal to 8° as a start, and then is divided by 2 until it reaches the arbitrary limit value of 0.5°. As the variables have no normalization of any kind, the function to maximize may have a peak along one of the variables, but of angular width much narrower than 0.5°. Such a case has not been taken into consideration for the results presented farther in this note ; it has no consequence on the Fisher direction but results in a possibly not totally achieved optimization.

A way out that does not involve a complex variable transformation could be to keep dividing the angle by 2 until all the variables have been used at least x times in the optimization. This has been tried on other data and did not appear to be satisfactory. A normalization of all the variables by their variance seems to provide better results, but this is still an ongoing study.

5.4 Determining the cut values

As has been written above, the value of a cut is determined as a function of the efficiency of this cut on the signal, through equation (6).

This efficiency is chosen such as the (absolute) number $N_{S_{out}}$ of signal candidates from the training sample which are removed by the cut is high enough to be insensitive to the statistical fluctuations, and low enough for the efficiency of the cut to be high – so as to stay in a local description of the distributions.

In order to know better the landmarks between which $N_{S_{out}}$ can be chosen, we have done a study with only one LDA cut, on the Ξ. Two observables have been chosen, in order to define the direction $\overrightarrow{u}$ simply by an angle. The number of background candidates that are filtered out can then be plotted as a function of this angle, and this plot can be drawn for various values of the cut efficiency on the signal, i.e. for various values of $N_{S_{out}}$.

As expected, it appears that when $N_{S_{out}}$ is too low, fluctuations appear in the curve and finding the maximum becomes difficult. Moreover, the maximum would only be representative of this training sample. When $N_{S_{out}}$ is too high, the maximum is very well defined, but the angle corresponding to this maximum is very close to the angle found by the Fisher criterion, and the gain in amount of background cut is low : in such a case, $N_{S_{out}}$ is a large enough proportion of the total amount of signal to give the optimized criterion a “tendency of being global”.

In this 2-dimension study, a reasonable value of $N_{S_{out}}$ was 200, but the number used in what follows ranges from 500 to 1500, with 25 dimensions, for the training candidates of the zone that is cut to be statistically representative of the actual population that is cut.

5.5 Statistics necessary

5.5.1 Evaluation of the statistics necessary

The first method which comes to mind to evaluate the statistics needed in the training samples, as well as to do systematic studies dealing with the determination of the LDA directions, consists in checking that, under different conditions, similar directions are found.

This method, for mathematical reasons, can’t work : whichever the criterion used, an optimization is performed [1], consisting in determining the position of the global maximum of a surface in an n-dimension space (the coordinates of the LDA vector).

Firstly, because of the algorithm used, the maximum that is found is actually local. It may possibly be the same as the global maximum, but this isn’t guaranteed. It is therefore not impossible that, for quite similar starting configurations, the optimization falls into two different local maxima, in which case the two directions which we want to compare will be far one from another and can even have different performances.

Secondly, the determined local maximum isn’t necessarily a narrow and well-defined peak : it could be a large plateau covering a wide range of LDA directions. In such a case, two distinct starting configurations may lead to directions which are far apart – each corresponding to a little fluctuation in the plateau (possibly of statistical origin) reached by the optimization – but having yet the same performance.

Hence one shouldn’t look at the angular proximity of two directions to compare the relevance, the stability or the consistency of two configurations or methods, all the more so as using the multicut method strongly enhances the effects mentioned above, as the i^{th} direction depends on all the $i-1$ previous ones.

One solution – very heavy to set up, so we didn’t try – consists in defining a discriminancy criterion, and in using it to compare the methods or the configurations. In our case, discriminancy in its usual meaning is of little use, as we need to extract a class according to criteria that are known only after the analysis, and not discriminate two classes. It is therefore better to define a performance (a cost function), and such a variable can be calculated only after the analysis ; this is the cause of the heaviness of this solution. A good performance criterion is for example the statistical uncertainty on the number of Ξ integrated in $p_\perp$ and corrected for the efficiency, taken with the value of the last LDA cut which gives the smallest error bar.

[1] When the Fisher criterion is used, the optimization doesn’t appear in the program but is yet made : through the fact that the mathematical expression of its result is known.

Determining the minimal statistics needed for the calculation of the LDA directions can then be done by calculating the performance for various sizes of the training samples (it should then be watched that the statistics of the test sample itself is high enough). In theory, the performance should rise with the size of the training samples, and saturate when the latter reaches the minimal size necessary for a good determination of the LDA directions.

The size of the test sample is a smaller problem, firstly because what is compared here are performances (as opposed to directions ; all vectors which image is in a common plateau have the same performance), and secondly because real data are used, and they are more numerous than the simulated data. Using real data doesn't lead to a problem with identifying the noise and the signal as this is not required to calculate a performance : a simple counting is enough.

When the statistics of the simulated data is high enough, another solution can be considered : these data can be divided into a training sample and a test sample. Doing so avoids going through the whole analysis process but makes the definition of a performance be less obvious. An estimation of the performance has to be substituted to it, and could be obtained via the efficiency of the cuts on the test samples and the known ratio between the amount of signal and background in the real data for a given cut level (whichever, under the hypothesis that the efficiency obtained on the test sample matches the real efficiency).

A more lazy way to do than trying for various amounts of candidates in the training samples, but a priori as reliable, is to check that the performance (or its estimation) of the i^{th} LDA cut is higher than that of the i–1^{th} cut tightened beyond the cut value at which the i^{th} cut should begin to be applied. If it is not the case, it is a strong indication that the statistics used to determine the i^{th} direction was not sufficient.

5.5.2 Fisher criterion

Using the Fisher criterion allows to temper the arguments brought in the previous paragraph : Fisher being a global criterion, it is almost insensitive to the local properties of the distributions. The LDA directions determined with this criterion are therefore more stable with respect to a change in the method or of the statistics used.

This being said, comparing two directions remains perilous. However, an interesting consequence of Fisher's criterion being global is that the evolution of the direction of the LDA vector with the cut number can be utilized, when the multicut method is used.

The statistics removed by the cut number i will have a low influence on the determination of the next direction, it can therefore be expected that this latter is close to the previous one, modulo the plateau effect. This can be seen in the graphs that show the evolution of each of the n coordinates of the LDA vectors as a function of the number of the cut, presented in figure 12 for two coordinates.

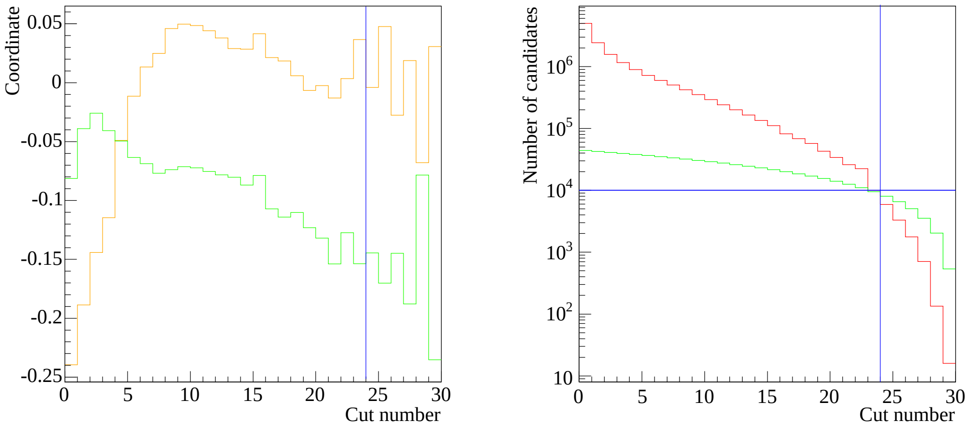


FIG. 12 – *Oscillations of the 21-coordinate* LDA *vector when the Fisher criterion is used : on the left, evolution of the 9th (in green) and 17th (in orange) coordinate of each of the 30* LDA *vectors calculated (for clarity, the 9th coordinate has been multiplied by 20 and shifted down by 0.15). On the right : evolution of the number of signal (green) and background (red) candidates used to calculate each of the 30 directions. The blue line materializes the beginning of the oscillations (left hand plot), and corresponds to* 10 000 *candidates (right hand plot).*

These graphs show that the direction evolves smoothly until some cut number, from which oscillations begin to show up, and then grow. This particular cut has therefore been determined with training samples which size is equal to the minimal statistics needed to calculate correctly the LDA vector. The number of candidates of the least populated sample (the background sample) at this stage is about 10 000. A look at figure 13 gives 11 000, and other studies all give a number between 10 000 and 12 000 for the beginning of the oscillations. Hence this will be considered as the lower limit for the statistics of either training samples, signal and background.

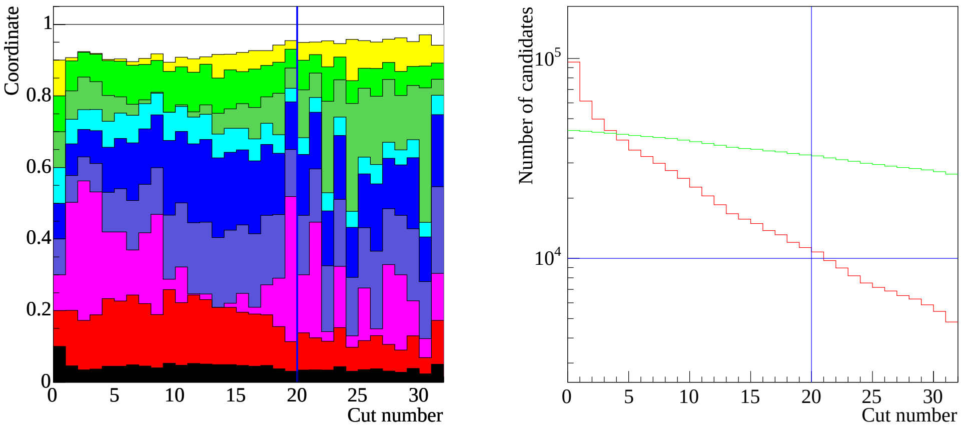


FIG. 13 – *Similar study to that of figure 12, with 10 coordinates. The left hand figure is this time the square of each coordinate (the sum equals 1 because the* LDA *vector is normalized), after a normalization of the 10 axis such as the coordinates of the first* LDA *vector are all equal to* $1/\sqrt{10}$

This limit has been determined with $n = 10$ and $n = 21$ dimensions and seems independent of the number of observables, which is a consequence of the fact that the distributions are projected on a 1-dimension sub-space (the LDA axis) for the calculation of the direction.

It can be objected that the oscillations may reflect the fact that the statistics cut at this stage becomes a significant proportion of the total statistics, and that Fisher hence becomes sensitive to the part that is cut, leading to the behavior illustrated by figure 14 for the optimized criterion. However, a still preliminary study done with the "lazy method" described at the end of § 5.5.1, applied to other data, seems to confirm that a minimal amount of candidates in the training samples is in the range of 7 000 to 10 000.

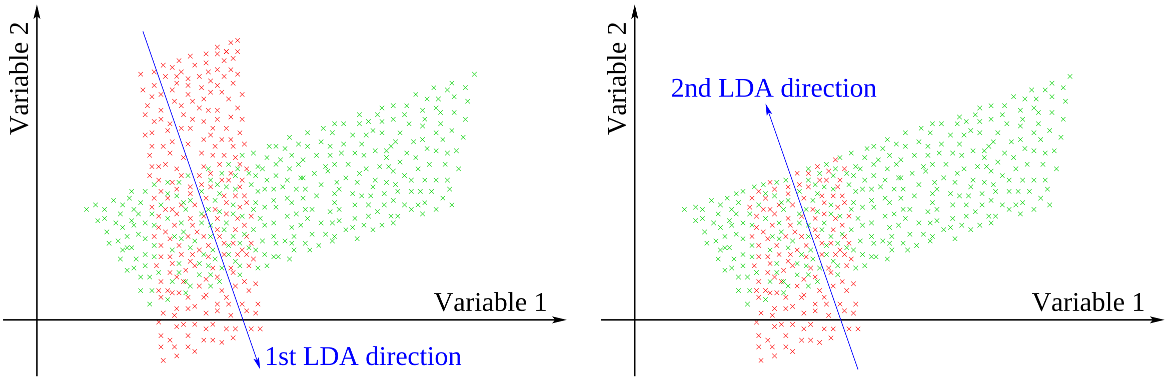


FIG. 14 – *Oscillations of the* LDA *vector when an optimized criterion is used : first (left) and second (right) directions given by the multicut method*

5.5.3 Optimized criterion

All this is unfortunately not valid when an optimized criterion is used : as shows figure 14 on the page before, the direction will naturally show strong oscillations with the multicut method. These oscillations are of course normal, but they make it impossible to use the method presented in the previous paragraph to determine an order of magnitude of the statistics needed.

So there is no simple mean of knowing this minimal statistics in the case of an optimized criterion. The solution consists in using the maximal statistics available, and to reduce the number of dimensions used in the space in which the LDA directions are calculated (or to do a study presented in § 5.5.1).

Surprisingly, a still preliminary study done with other data by using the "lazy method" of § 5.5.1 indicates that only 2 000 candidates would be enough to determine correctly an optimized direction.

As an example, in the study done on the multi-strange baryons, around 70 000 signal candidates and about 2 000 000 background candidates have been used in the training samples for the first direction. The last direction used in the analysis has been determined with 50 000 signal and 33 000 noise for the Ξ, 26 000 signal and only 7 000 noise for the Ω. More details can be found in chapter 5 of [4].

5.5.4 Possible and impossible solutions

Various solutions exist to reduce the number of dimensions of this space, but not all of them are realizable :

- calculation of the discriminancy as a function of the number of directions ;
- under-optimal LDA ;
- Principal Component Analysis.

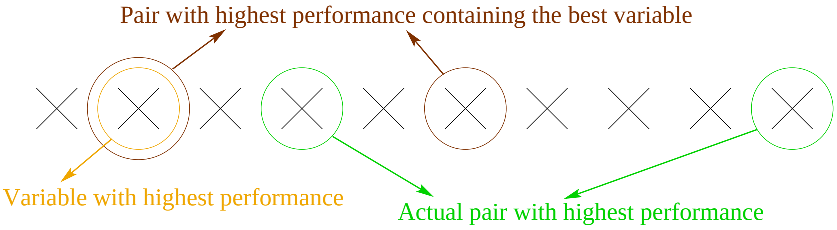

Pair with highest performance containing the best variable
Variable with highest performance
Actual pair with highest performance

FIG. 15 – *Drawback of under-optimal LDA : the black crosses symbolize the n variables that are usable in LDA ; under-optimal LDA provides the most performant pair of directions (in brown) among the pairs which contain the most performant variable (in orange), but the actual most performant pair (in green) actually doesn't contain the most performant variable*

Calculation of the discriminancy with respect to the number of directions : it consists in determining the value of a "performance criterion" of the cuts obtained (this is traditionally the discriminancy, but, as we have seen, in our case the definition of a performance downstream the analysis is needed) as a function of the number of variables used in the LDA. For each number of variables $j \leqslant n$, there exists a j-uplet which gives the best performance P_j. Plotting P_j as a function of j gives a monotonically rising curve[1] which maximal value is P_n, and which "derivative" tends towards 0 when j tends towards n (*cf* the green curve in FIG. 16). A value m of j can therefore be defined such as P_m is close enough to P_n, m being however significantly lower than n.

So using the corresponding m-uplet gives cuts which performance is close to the maximal performance, with yet a space of smaller dimension. The main drawback of this method is that it requires the test of all the combinations of directions, i.e. $2^n - 1$. For some applications, an automatic program could possibly be used for $n = 10$ (1023 combinations), or with limiting hypothesis on m such as $8 < m < 15$, but for most of the applications n is a few tens[2], so this method is actually never used.

[1] This is the case when the statistics of the training samples is high enough. This isn't our case and is precisely the reason why reducing the number of dimensions is desirable, it is thus possible to obtain a curve which has a maximum beyond which the number of dimensions is too high for the available statistics to provide the optimal LDA direction.

[2] In our case : $n = 25$ gives 33 million of combinations.

A partial solution is under-optimal LDA : this recursive method consists in searching the most performant j-uplet only among those which $j-1$ directions are those of the most performant $j-1$-uplet. So the variable giving the most performant cut is first searched, then the pair of variables which gives the most performant LDA cut is searched among the pairs that contain the previously selected variable, and so on. The number of combinations to test is then reduced down to $\sum_{j=1}^{n} j = \frac{n(n+1)}{2}$, which still makes 325 combinations for 25 dimensions... i.e. a whole year of calculation on the farm for the present analysis and for a performance defined downstream the analysis.

As illustrated by figures 15 and 16, the main drawback of this method is that, generally speaking, the most performant j-uplet doesn't contain the most performant $j-1$-uplet, hence the name of "under-optimal LDA". However, this solution is often used, sometimes in combination with the first method presented : an exhaustive search is made for e.g. 5 variables, and under-optimal LDA is started from a search of the 6-uplets containing the absolute most performant 5-uplet.

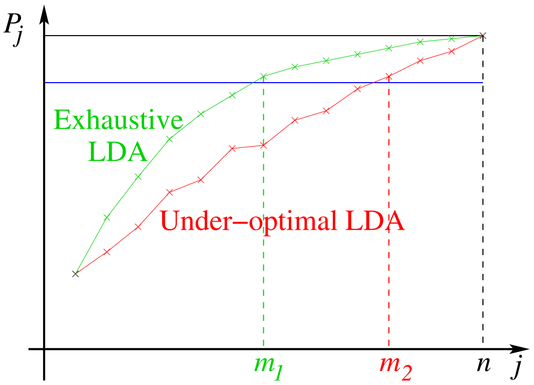


FIG. 16 – LDA *performance P_j as a function of the number j of variables used. A satisfactory performance (in blue) is reached faster by an exhaustive test of all the combinations (m_1 variables used, in green) than by under-optimal* LDA *($m_2 > m_1$ variables used, in red)*

The Principal Component Analysis (PCA) is a matrix-based analysis (based on a diagonalisation, so it is simple and fast) which gives for a distribution an ordered base of the space $(\overrightarrow{v_1}, \overrightarrow{v_2}, \cdots, \overrightarrow{v_n})$ which first vector $\overrightarrow{v_1}$ drives the direction along which the distribution has most information, and so on for the other vectors. The proportion of information along one of the vectors $\overrightarrow{v_j}$ is given by the square of the corresponding eigenvalue λ_j. It is therefore possible to compress data, storing the m first coordinates of each observation corresponding to a compression of $100\left(1 - \sum_{j=1}^{m} \lambda_j^2\right)$ %.

In our case, the PCA would be used to reduce the number of variables used in LDA, by getting rid of the directions which don't carry enough information. These latter can change at each step, so the PCA needs to be applied in the n-dimension space after each LDA calculation (and cut), determined in a m_i dimension sub-space.

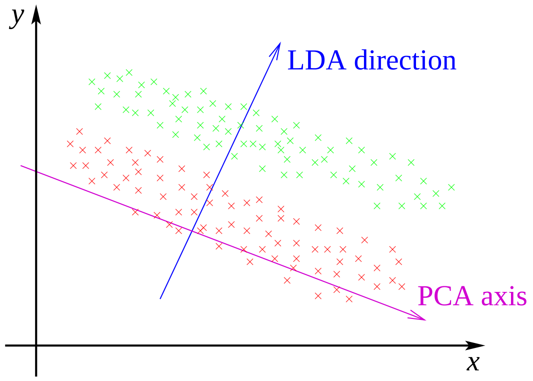
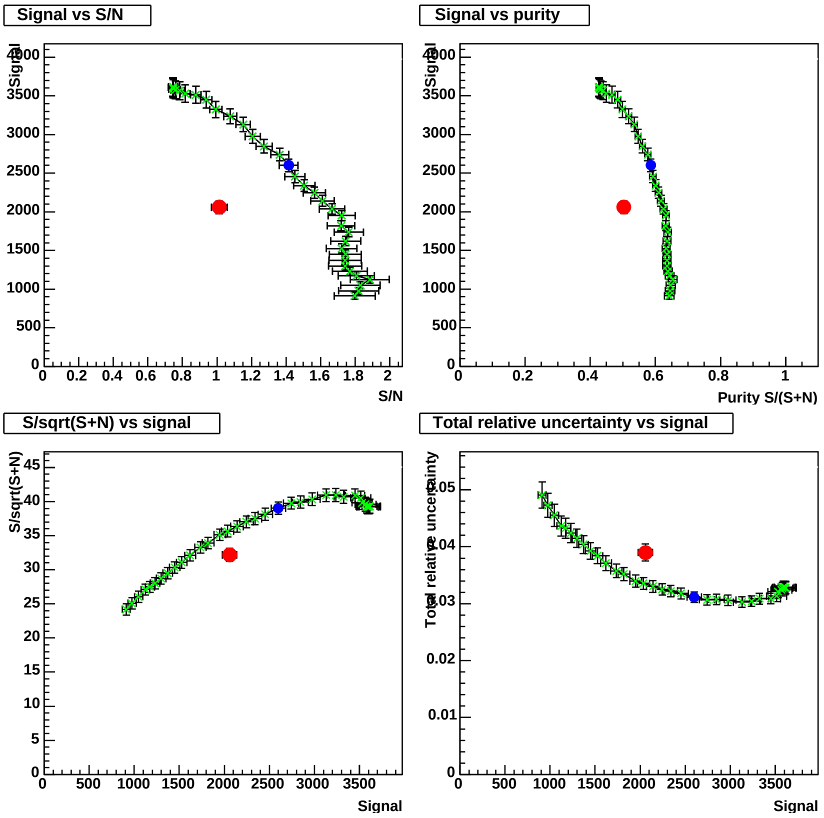


FIG. 17 – *Least informative direction, being yet the most discriminant*

The main problem with this method is that the PCA is *not* a pattern classification method, it therefore doesn't deal with several distributions. So for this application, the signal and background distributions need to be mixed (with equal statistics, except if more weight is wanted for one of the classes) before applying the PCA. This in itself is not a problem, but the LDA direction determined in n dimensions (whole space) is not guaranteed not to be orthogonal to the first PCA direction (case of parallel and long signal and background distributions, illustrated in figure 17), in which case the directions removed by the PCA are precisely those which have the best discriminancy... There is no mathematical criterion to evaluate a priori the improvement brought by the PCA ; the only solution is to try with and without the PCA, and then to compare. The fact that the PCA is simple and fast to set up makes it preferable to under-optimal LDA, but one method doesn't replace the other : under-optimal LDA may help where the PCA doesn't bring any improvement.

The PCA can also be used to solve the problem of Fisher criterion's denominator going to 0 : correlations between variables can indeed be reduced, by a process called *whitening*, which consists in calculating the PCA matrix of the distributions, and then to normalize the distributions along the directions of the PCA base by a factor of $\frac{1}{\lambda_j}$, λ_j being the eigenvalue associated to the j^{th} vector of the PCA base. In our case however, the linear correlations can't be completely removed, as both distributions (signal and background) are simultaneously renormalized by the PCA, and their correlations are not necessarily identical. It is yet a nice tool to apply the optimized LDA in a normalized base, i.e. with all variables showing values of the same order of magnitude.



Signal S with respect to the signal to noise ratio S/N (left)

Signal S with respect to the purity $\frac{S}{S+N}$ which absolute maximum is 1 : efficiency-purity diagram (right)

Significance $\frac{S}{\sqrt{S+N}}$ with respect to the amount of signal S (left)

Relative error $\frac{\sqrt{S+N+k^2 N_{estim}}}{S}$ with respect to the amount of signal S (right)

FIG. 18 – *Diagrams allowing to evaluate the performance of the cuts for the Ω. The red dot is for the classical cuts ; the green crosses materialize the curve followed when changing the last* LDA *cut. The blue dot is the* LDA *cut value which gives the same background level as the classical cuts*

6 How to choose the last LDA cut value

The last LDA cut value is determined in a different way than the others : the variable used to choose it is neither the efficiency nor the signal to noise ratio, but the relative uncertainty. The latter is indeed directly related to the statistical error bar on the final result. For this error bar to be as small as possible, the last LDA cut needs to be varied until the value that leads to the smaller relative uncertainty is found.

It is noteworthy that LDA provides such an easy search of this extremum : only one parameter needs to be varied : the value of the last cut (all the previous values are optimal). With the classical cuts, this extremum is searched by varying as many parameters as there are cut variables, i.e. a dozen.

The search is all the easier with LDA as the variation of the relative error with respect to the cut value is a function that is piecewise monotonic : when the cut is tightened with respect to the optimum, the amount of signal drops faster than the error bar because the amount of background is low. When the cut is loosened with respect to the optimum, the amount of background rises faster than that of signal, and the latter rises slowlier than the error bar. As a consequence, a valley-shaped curve is obtained and the minimum is easy to find.

The phenomenon is illustrated by figure 18 for several variables : signal to noise ratio, purity, significance[1] and relative uncertainty[2], all as a function of the amount of signal (or efficiency). The locus drawn in each of these diagrams by a loosening or a tightening of the last LDA cut is shown in green, to be compared to the red dot which shows the “performance” obtained with the classical cuts.

The amount of signal being proportional to the efficiency, these two variables are strictly equivalent, but

[1] Defined here as $\frac{S}{\sqrt{S+N}}$.

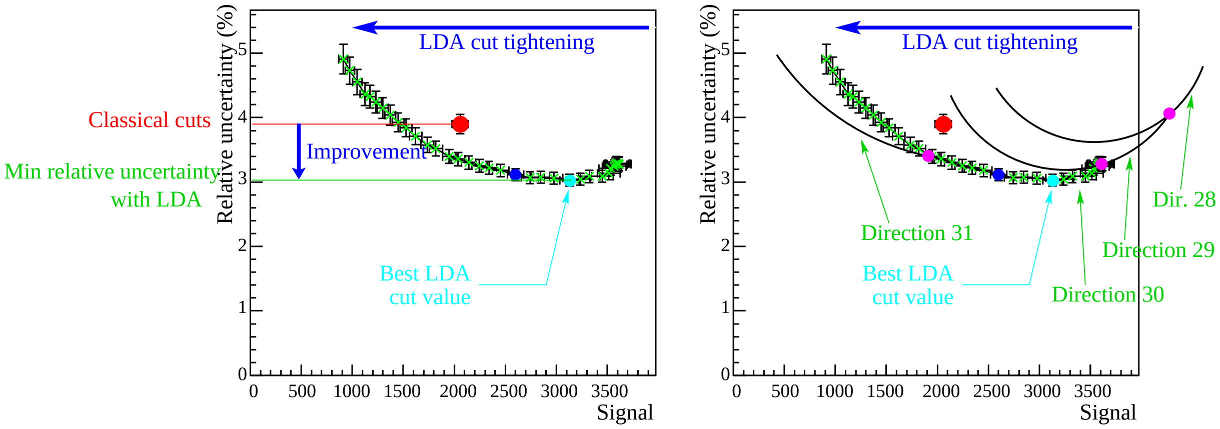
[2] Equal to $\frac{\sqrt{S+N+k^2 N_{estim}}}{S}$, with k the scaling factor used to have the estimated background match the real background.

the simulation has not been used yet at this stage, so the efficiency remains unknown. This is the reason why the amount of signal is chosen as a variable. The variation of this amount of signal gives the direction of the variation of the LDA cut (the efficiency drops when the cut is tightened).

The method to choose the number of LDA cuts to apply is illustrated in figure 19 : in the right-hand plot, the magenta points show the position of the cut along the penultimate and the ante-penultimate directions, which are determined by the program ; i.e. at the magenta point located at the intersection of the curves "direction 28" and "direction 29" for example, the algorithm has determined a new LDA direction (the 29th) whose performance is higher than that of the previous direction. The envelope of these valley-shaped curves is the locus drawn when the LDA cut is gradually tightened (and directions progressively added) ; this locus has a minimum, which is the minimum of the curve obtained with the searched number of directions (30 on the figure). The optimal number of cuts to use in this case is therefore 30. In practice, one can determine visually on an invariant mass plot the number of LDA cuts that give the same amount of background (under the peak) as the classical cuts, can then determine the value of the minimum for the neighbouring numbers of cuts, and then compares the values to obtain directly the optimal number of cuts[1].

Determining the number of directions to use by drawing the envelope of all the valley-shaped curves can be done in only one pass on the data (possibly even just on a subset), so it is not at all necessary to have classical cuts at disposal to start with : LDA is a totally stand-alone method.

This searched minimum is perfectly visible (bottom-right subfigure in FIG. 18, relative uncertainty as a function of the amount of signal). Yet, the final value of the LDA cut is not determined at this stage, because the error bar due to the efficiency correction (or any other further "manipulation" of the data) is not taken into account in this relative error. The graphs shown here just give an idea of the zone in which the LDA cut has to be varied.



(a) The value chosen for the last LDA cut is that which gives the smallest relative uncertainty. The gain brought by LDA with respect to the classical cuts can be read directly on the graph

(b) Each LDA direction gives a valley curve ; the choice of the number of cuts is made by considering the minimum of each valley : the lowest one is obtained with the suitable number of LDA directions

FIG. 19 – *Method for tuning the last* LDA *cut ; the amount of signal is bijectively linked with the* LDA *cuts tightening*

It furthermore has to be noticed that those diagrams have been built using all the candidates (integrated in $p_\perp$). Given that the improvement provided by LDA is strongly $p_\perp$-dependent (*cf.* § 7), the relative uncertainty on a variable, such as the corrected production yield for example, will reach its minimum for an LDA cut value for which the minimum of the (raw, uncorrected) signal relative uncertainty will be minimal only in some restricted range in $p_\perp$. The value of the LDA cuts therefore has to be chosen for each considered physical variable. In practice however, these values are close one to another and a common LDA cut value can be found so as to be close to the absolute minimum for several physical observables simultaneously, at least in the analysis given as an example here.

[1] If the statistics in the training samples is not sufficient to calculate enough LDA directions, the number of directions to use should be the maximal number allowed by the statistics of the training samples, which, as already said, is the case for the Ω analysis shown here, but not for the Ξ analysis.

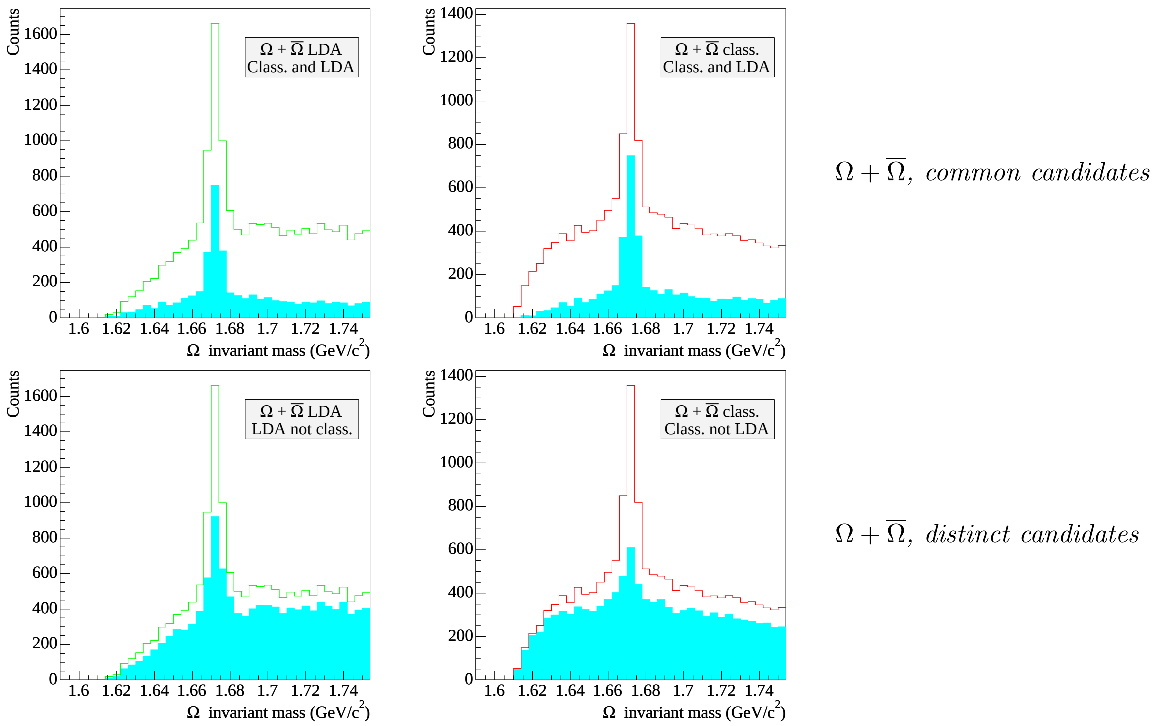


FIG. 20 – *Invariant mass distribution of the* $\Omega + \overline{\Omega}$ *candidates after* LDA *(in green) and classical (in red) cuts. Top row, cyan : distribution of the candidates selected by both sets. Bottom left, cyan : candidates selected only by the* LDA *cuts. Bottom right, cyan : candidates selected only by the classical cuts*

7 Some results

7.1 Candidates selected by both sets of cuts

Figure 20 shows, for the Ω, the proportion of signal and background selected by both the classical and LDA sets of cuts, or, on the contrary, selected only by one of the two sets of cuts. For the signal, almost half of the Ω selected by the LDA cuts have not been selected by the classical cuts, which means that a significant part of the selected populations is not shared by both sets of cuts. If the results obtained later on are similar for both methods, they are all the more reliable as the part of the phase space selected by both sets is small.

As for the background, in both cases more than 75 % of it are candidates that haven't been selected by the other set of cuts, which also guarantees reliable results if they are similar, from the evaluation of two different backgrounds.

7.2 Characteristics of the LDA cuts

A possibility to check that no classical set of cuts is close to the LDA cuts found is the check that the distributions of the various variables after the LDA cuts are not cut steeply.

A more subtle way to observe the behavior of the LDA cuts uses the definition of a steepness criterion, i.e. a variable Q which is equal to 0 when the cut is uniform, and to 1 when it is as steep as a classical cut.

Let f be the distribution of a cut variable x on the limited domain $[0; X]$ before the considered cut is applied (for example $x < x_{cut}$), and g the distribution after this cut, which has an efficiency ε. The criterion Q sought satisfies among other things :

- uniform cut $\implies \dfrac{g}{f} = cst \implies Q = 0$;
- steep cut $\implies \dfrac{g}{f} = 1$ for $x < x_{cut}$, $\dfrac{g}{f} = 0$ for $x \geqslant x_{cut} \implies Q = 1$.

Let $\mathscr{F} = \dfrac{\int_0^x f(u)du}{\int_0^X f(u)du}$ be the normalized primitive of f, $\mathscr{F}^{-1}$ its inverse function:

$$\begin{aligned}\mathscr{F}^{-1} \quad : \quad & [0;1] \longrightarrow [0;X] \\ & \frac{\int_0^x f(u)du}{\int_0^X f(u)du} \longmapsto x\end{aligned}$$

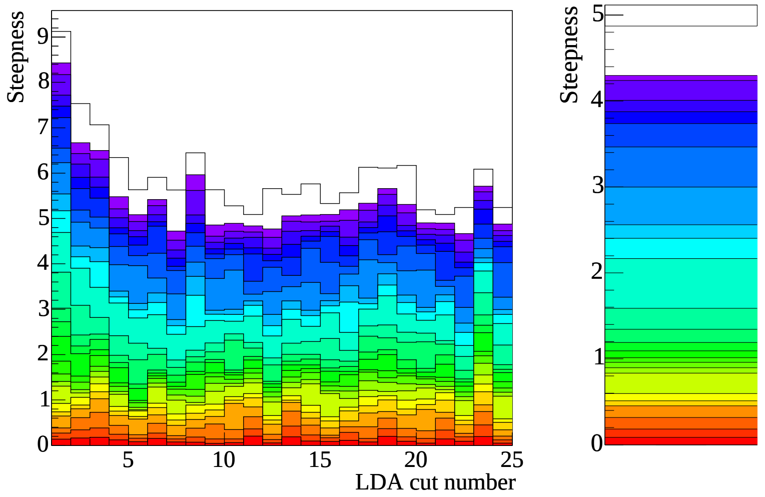
and h the function defined by $h = \left(\dfrac{g}{f}\right) \circ \mathscr{F}^{-1}$:

$$\begin{aligned}h \quad : \quad & [0;1] \longrightarrow [0;1] \\ & x \longmapsto \frac{g}{f}\left(\mathscr{F}^{-1}(x)\right)\end{aligned}$$

By construction, h draws the ratio g/f as a function of the normalized integral to x under f. The schemes of figure 24 on page 27 give the shape of h for various cuts. It appears that their steepness can be determined directly from $|h - \varepsilon|$; a further normalization by the integral $(2\varepsilon(1-\varepsilon))$ then leads to the formula of the searched criterion:

$$Q = \frac{1}{2\varepsilon(1-\varepsilon)} \int_0^1 |h - \varepsilon| \tag{9}$$

When the steepness Q_j of an LDA cut is very close either to 0 or to 1 for each variable j (for both the signal and the background), it is possible to have a classical set of cuts equivalent to the LDA cut applied. When only some of the steepness factors are equal to 0 or 1, there is only a low probability that the corresponding set of classical cuts give identical distributions to those obtained with LDA for the variables for which the cut steepness lies between 0 and 1. Figures 21 and 22 on the following page show that this is anyway not the case in the Ξ analysis, for any of the LDA cuts used. The situation is similar for the Ω.



(a) For each of the 24 LDA cuts

(b) For all the LDA cuts together

FIG. 21 – *Steepness factor of the* LDA *cuts on the* $\Xi + \overline{\Xi}$ *signal: the steepness factors* Q_j *of each variable* j *have been added and assigned a color each*

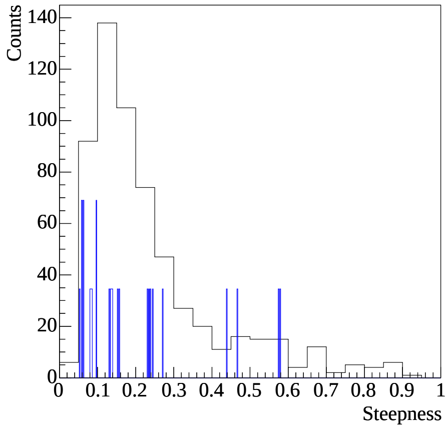


Fig. 22 – *Distribution of the steepness of the Ξ* LDA *cuts on the 25 variables, in black. The blue distribution is that of the steepness of all the* LDA *cuts together on the 25 variables (it has been normalized for clarity)*

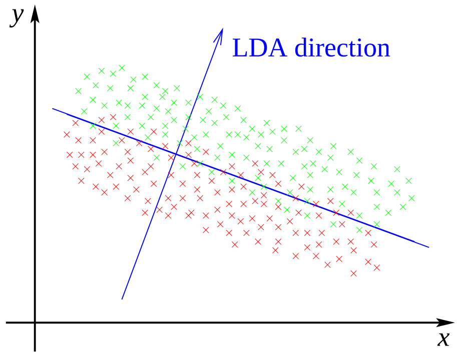


Fig. 23 – *Example of 2-dimensional distributions which lead to a null steepness of the* LDA *cut along both coordinates. The cut hyperplane is the blue line that is perpendicular to the* LDA *direction*

An important remark is that the fact that the steepness of an LDA cut for a variable is zero doesn't imply that this variable is useless in the calculation of the LDA direction, as proved by the example of figure 23 : apart from the extremities of the distribution (though it is possible to find distributions which also pass this criterion), the background and signal are uniformly cut in x and y ; the steepness of this cut is therefore zero for both variables, while the LDA direction is nothing but a linear combination of them. So the steepness factor is useless for the problem mentioned in paragraph 5.5 concerning the lowering of the number of variables used in LDA.

7.3 Improvement brought

As for the previously given Ξ and Ω results, the plots presented here have been obtained on the central collisions of the year 2001 Au-Au 200 *GeV* data. Details can be obtained in chapters 5 and 6 of [4].

Due to a bug in the LDA code discovered after the analysis was completed, 4 of the 25 variables used in Fisher were actually not used in the optimization : the $\cos\theta^*$ and the number of hits of the three daughter tracks in the TPC. This has no influence on the physical results, but it means that the actual improvement that LDA can bring is higher than shown here.

Figure 25 on page 28 shows the invariant mass distributions obtained for the Ξ and for the Ω with both sets of cuts, LDA being tuned for the background under the peak to be at the same level as that obtained with the classical cuts. An improvement in raw number of signal is clearly visible for both particles. The improvement also appears to be higher for the Ω than for the Ξ : this is a consequence of the fact that the classical analysis cuts are much tighter for the Ω (the efficiency of those that are replaced by the LDA cuts is around 20 %) than for the Ξ (about 45 %), which means that the “remaining space” for an improvement is much lower for the Ξ (factor of 2) than for the Ω (factor of 5).

Although the optimization is not based on this variable, the behavior of the LDA improvement factor $\mathcal{F}$ as a function of $p_\perp$ is interesting to look at. It is defined as follows :

$$\mathcal{F} = \frac{S_A - S_C}{S_C} \tag{10}$$

where S_C and S_A are the uncorrected amounts of signal obtained respectively with the classical and LDA cuts.

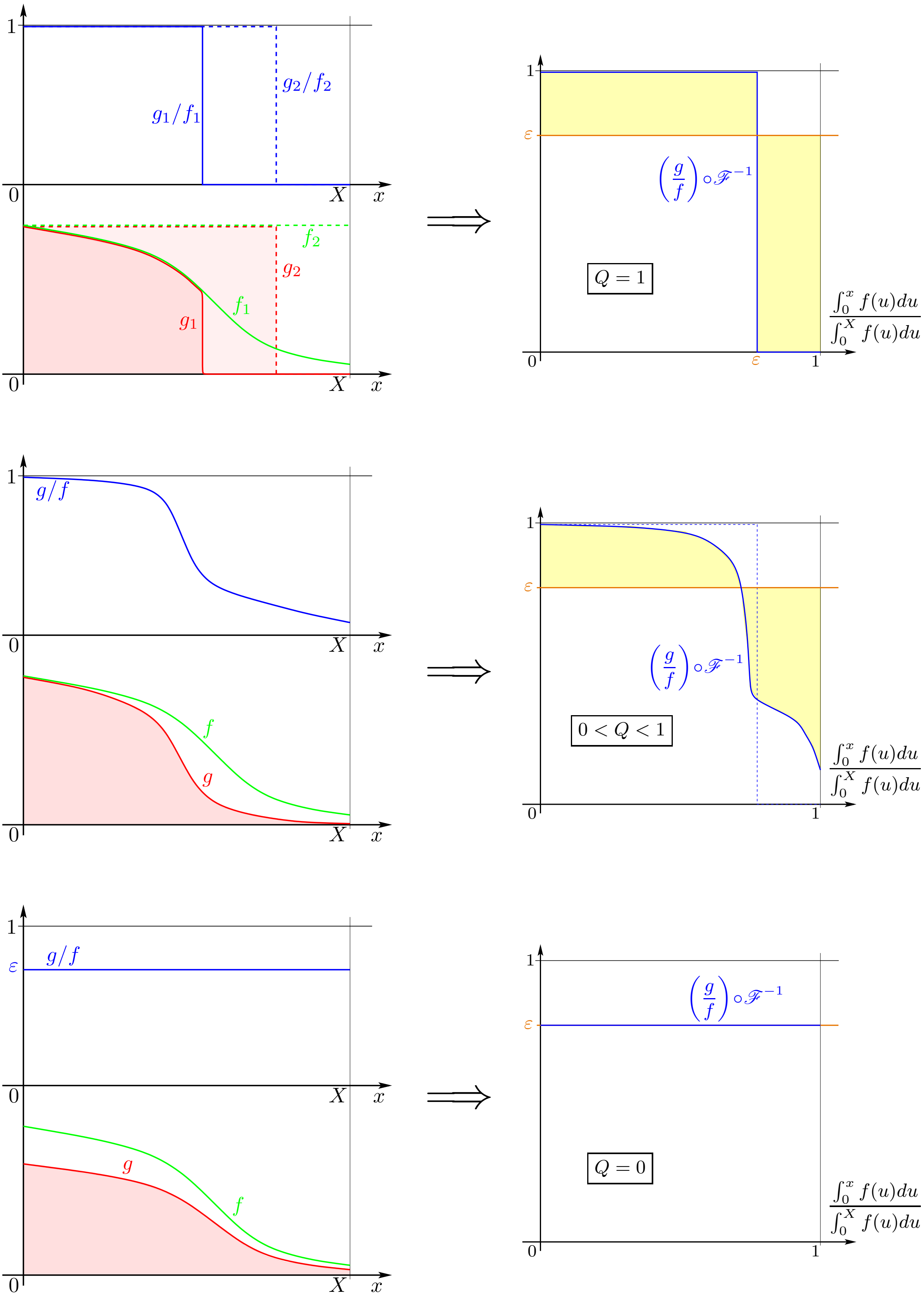
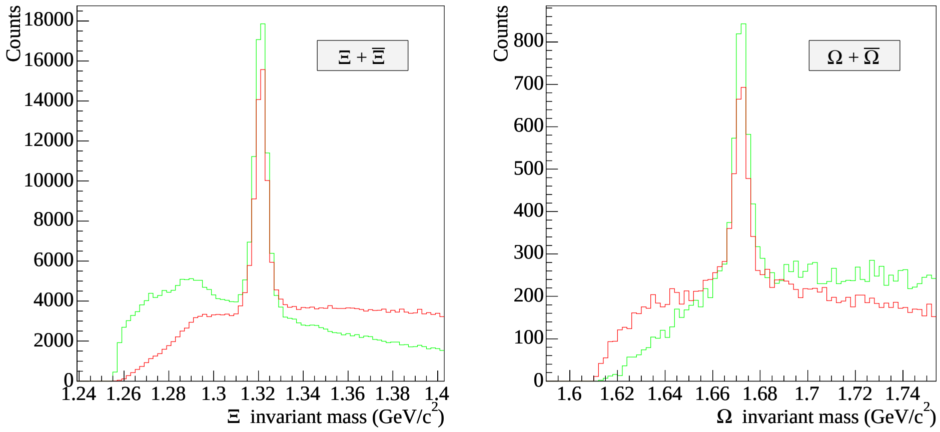


FIG. 24 – *Calculation of the steepness factor of a cut: for a steep (classical) cut on the top row, for an intermediate (*LDA*) cut in the middle, and for a uniform cut in the bottom row. The distribution of the cut variable is in green before the cut is applied (f), in red after it is applied (g). The ratio g/f is plotted in blue in the left-hand figures. On the right is shown the function h in blue, as well as the variable Q (steepness of the cut applied) calculated from a normalization of the yellow area*



(a) For the $\Xi + \overline{\Xi}$

(b) For the $\Omega + \overline{\Omega}$

FIG. 25 – $p_\perp$*-integrated invariant mass distributions with the classical (in red) and* LDA *(in green) cuts*

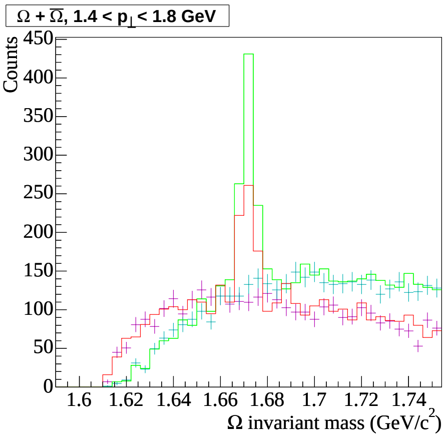


FIG. 26 – $\Omega + \overline{\Omega}$ *invariant mass distributions for the* $p_\perp$ *bin* $1.4 < p_\perp < 1.8$ *GeV/c with the classical (red) and* LDA *(green) cuts (the magenta and cyan error bars show the background level estimated with the rotating method)*

The improvement brought by LDA for the Ω in the $p_\perp$ bin $1.4 < p_\perp < 1.8$ GeV/c is shown in figure 26 : it is obvious that the improvement is higher than that integrated in $p_\perp$, as one can see by comparing with figure 25, which means that the improvement is $p_\perp$-dependent.

The variation of $\mathcal{F}$ as a function of $p_\perp$ is presented in figure 27. For the Ξ as for the Ω, a peak at mid-$p_\perp$ is observed. This is a consequence of the use of candidates unselected in $p_\perp$ in the LDA training samples : the calculated LDA directions have logically favored the candidates of the $p_\perp$ zone where they are most numerous, i.e. those in the mean-$p_\perp$'s.

This result is interesting and promising, as it confirms that it is possible to create LDA sets of cuts that are specific to some $p_\perp$ regions which suffer a low statistics ($p_\perp < 1$ GeV/c and $p_\perp > 3$ GeV/c in this analysis), by using in the training samples only candidates that belong to those zones.

In the case of the Ξ, the improvement factor seems to rise again at high-$p_\perp$: this is not explained and could result from the shape of the Ξ distribution in the phase space.

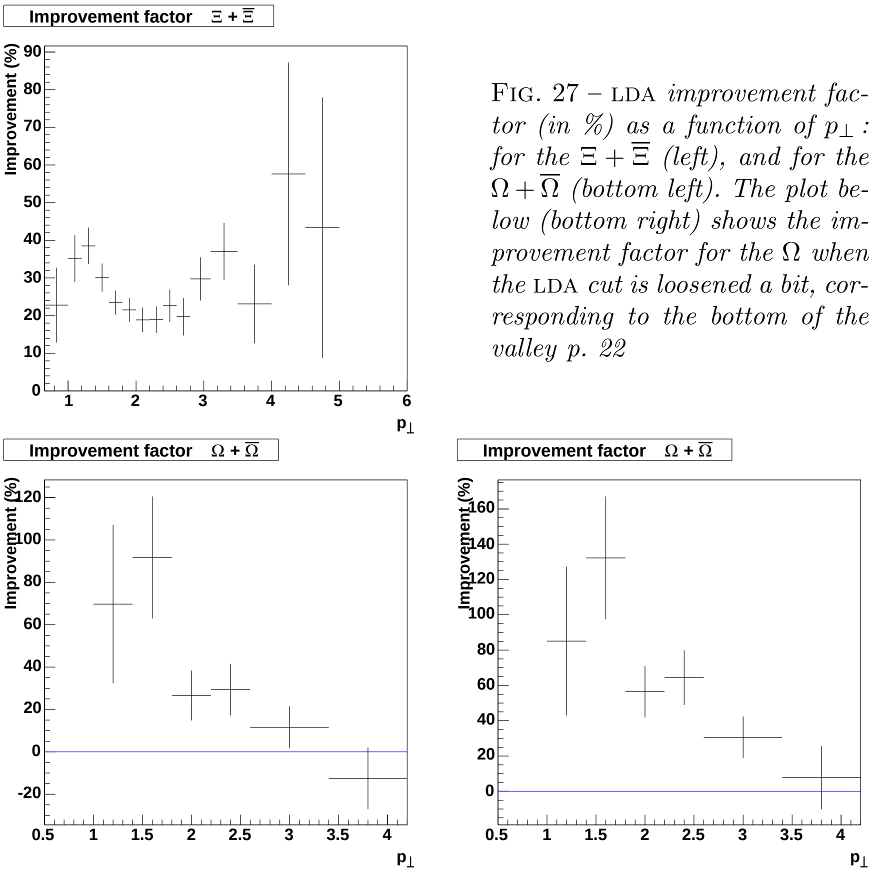


Fig. 27 – LDA *improvement factor (in %) as a function of $p_\perp$: for the $\Xi + \overline{\Xi}$ (left), and for the $\Omega + \overline{\Omega}$ (bottom left). The plot below (bottom right) shows the improvement factor for the Ω when the* LDA *cut is loosened a bit, corresponding to the bottom of the valley p. 22*

Conclusions

From a multi-variable pattern classification method, the Fisher linear discriminant analysis, we have developed a method of selection of the signal among a high amount of background : the multicut-LDA with optimized criterion, which appears more performant, simpler and more handy than the classical selection method.

The results obtained on the Ξ show a good agreement between the classical and LDA methods [4]. The Ω analysis also shows that both methods agree with each other, but the large statistics uncertainty makes it a weaker proof as the Ξ. Below follows a list of the various advantages provided by LDA (there is no list of drawbacks as I haven't found any yet ;-)).

Comparison with the classical cuts and the neural networks

The table on the next page shows some characteristics of the two mainly used methods – the classical cuts and the neural networks – and of LDA. The words emphasized in bold (and green) are the positive characteristics.

The difference between the classical cuts and LDA is obvious, but the preference of LDA over the neural networks may require additional explanations : the choice of LDA over a pattern classification method like the neural networks is justified by its simplicity.

Neural networks reach, in theory, a higher performance. But in practice, their non-linearity, the problem of overtraining (which doesn't exist with LDA, the locality degree of description of the distributions is managed in a trivial way), the choice of the number of hidden layers and of neurons make them difficult to use and make their training be long and tedious. A bad choice of their configuration or a not careful enough training rapidly result in lower performances than what could be expected. Neural networks also suffer from the huge background statistics: they focus on removing its overwhelming part and leave untouched the comparatively small amount of background which is close to the signal area. This problem can be avoided by cascading several neural networks (it can be seen as an equivalent of multicut-LDA), each stepping up the S/N ratio by an order of magnitude, but at the cost of an exploding number of parameters of the method.

	Classical cuts	**Multicut LDA**	Neural networks
• Optimized	No	**Yes**	**Yes**
• Linear	**Yes**	**Yes**	No
• Setting up	**Trivial, fast**	**Easy, fast**	Complex, long (Choose # layers and neurons)
• Training	**None** or long and complex	**Simple**	Complex Overtraining
• Final tuning	Complex, long	**Simple, fast**	**Simple, fast**
• Parameters	**Few**	**Few**	Many
• Clarity	**Under control**	**Under control**	"Black box"
• Boudary shape	Linear	**Linear, but** **multicut ⇒ OK**	**Non linear**
• Selected volume	Connex	Connex	**Non connex**

The two advantages of a neural network over LDA are its non-linearity (which is also a drawback) and its ability to select as being signal a zone of the phase space which may not only not be convex, but also not even connex. The first advantage disappears with the developments of LDA that we have done, thanks to the application of several LDA cuts. Remains the second one, from which no large improvement should be expected, firstly for the reasons mentioned above, and secondly because it is rather unlikely that the zone in which the signal lies is not connex or has a sizable concavity.

Lowering of the size of the statistical error bars

The gain in relative uncertainty brought by LDA is 20 to 30 % for the Ω (depending whether the variable looked at is the inverse slope or the production yield at mid-rapidity). For reasons given earlier, the improvement is smaller for the Ξ (around 20 %) but a study showed that replacing also the reconstruction cuts with LDA cuts rised the Ξ raw yield improvement from $\simeq 20$ to $\simeq 45$ % (for an equal amount of background) [6]. As a consequence, a 30 % drop of the relative uncertainty on the Ξ production yield could also be expected, but we didn't think this was worth a replacement of the reconstruction cuts and a re-production of the strangeness data.

The improvement is therefore not sufficient to be able to measure new observables. Yet, the statistical error bars are sensibly reduced. It also has to be kept in mind that this improvement has been obtained over *optimized* classical cuts.

For other centralities, other energies and other collision systems, the classical cuts are not optimized and it has been shown that LDA provides quicker and better results than the classical method, with an improvement reaching for example 40 % for the Ξ raw yield with though classical reconstruction cuts [6, 7], which results in a wider possible coverage in $p_\perp$ (this was already foreseeable from the central Au-Au 200 *GeV* analysis results, although not presented in [4], as LDA provides a usable Ξ signal in the bin $0.5 < p_\perp < 0.7$ *GeV*, while the classical cuts don't). The development of an optimizable (and optimized) and performant analysis method was all the more necessary as the future runs will often be scans in energy or in atomic specie, which deliver a smaller amount of data than the long 200 *GeV* p-p and Au-Au runs and offer little hope for extra data from further similar runs. Accurate measurements of the interesting observables in these "smaller" runs therefore require methods which extract the maximum out of the data.

Fast and simple cut optimization

LDA, by providing an optimized transformation from the n-dimension space of the variables used as cuts to a 1-dimension subspace, makes cut-tuning be extremely fast, as it just consists in finding the minimum of a

1-dimension function that is monotonic on both sides of its minimum. Thus the cut tuning is fast and simple.

Furthermore, a set of cuts determined on e.g. the central Au-Au 200 *GeV* data can be used extremely quickly on events of lower multiplicity, such as mid-central / mid-peripheral events [4] or 62 *GeV* collisions [7]. It simply consists in loosening the LDA cuts by finding the new minimum of the 1-dimension function. If the gap in multiplicity is not too important, the performance should not be too different than that given by an LDA set specially trained on such events.

Some observables though require an improvement in a specific domain. For example the production yields are very sensitive to the low transverse momenta – because it decreases exponentially and because the contribution of the extrapolated part in the total measured yield is very large for the multistrange baryons (more than 40 % for the Ω) –, the R_{AA} requires a spectrum in p-p collisions, etc... LDA helps there by giving quickly optimized cuts suited for the conditions wanted, as one trains it with candidates that lie in the portion of the phase space where the improvement is needed.

Internal LDA systematic error

Because LDA transforms the phase space into a 1-dimension subspace, it provides a natural systematic study that consists in tightening and loosening the LDA cut while keeping it optimal, thus avoiding all the efforts needed to find a second set of classical cuts that would be both efficient and different enough from the first one. This 1-dimension space offers a total control over the variation range around the nominal value in terms of efficiency and raw amount of signal.

To improve a systematic study, it is also fast and easy to find one or several other sets of LDA cuts determined by adding (or removing) classical cuts[1] or any other kind of selection. This may simply be LDA cuts determined in different $p_\perp$ ranges to get a better improvement in each region, and used over the whole spectrum for a systematic comparison, for example.

Classical/LDA systematic error

A systematic error can also be taken out of the comparison of the classical and LDA results, as both methods cut very differently in the phase space. If no classical cuts have been determined before the use of LDA, or determined but not optimized, this could be impractical because the statistical error on the classical results would be large. In such a case, it may be relevant to develop a simple method to derive a reasonably good set of classical cuts from an LDA set. We have not investigated this possibility (yet) ; it may be that there is no simple method and that, even though LDA would provide an estimate for a classical starting point (which already saves time), one would still need to tune by hand or to use a maximization algorithm in n dimensions.

Cut optimization with an internal tracker

Last but not least, LDA can be used for an easy cut optimization with the Silicon internal trackers (SVT and SSD). As the strange particles have a large $c\tau$, they can decay before and after any layer[2]. Hits in the most inner layers improve the accuracy of some cut variables, so different sets of cuts should be used. To take full advantage of the internal trackers, a classical method would require the creation of one set of cuts per configuration of the points in the Silicon layers. Assuming an efficiency of the detectors close to 100 %, a 3-particle decay with 4 layers of Silicon leads to 125 sets of cuts to be found !

LDA allows to use only one set of cuts (and optimized on top of that) for all configurations. The idea is to incorporate the configurations as LDA variables, for example the number of hits in the internal trackers for each daughter track[3]. So an optimal integration of the Silicon detectors in the cuts simply requires the addition of two (for Λ and K_s^0) or three (for Ξ and Ω) variables to LDA.

[1] It might sometimes be necessary to add some classical cut, for example to get rid of an observed or known bias that occurs in a certain zone of the phase space (often geometrical, which involves e.g. z, but also the decay lengths), or to remove explicitly a given configuration (reflections for example, or splitting via the number of hits in the TPC).

[2] A charmed meson would necessarily decay before the first layer, which simplifies the problem a lot because, the efficiency of the detectors being high, one may require that all layers, or all but one, have been hit by the daughters.

[3] With some weighting to take into account which layers (rather external or rather internal) have been hit, if we want to consider the case of layers with efficiency below 100 %.

Bibliography

[1] R. Duda, P. Hart, D. Stork
Pattern classification.
(Wiley Interscience)

[2] S. Faisan : private communication

[3] P. Lutz
Un exemple d'analyse multidimensionnelle : l'analyse discriminante.
Cours de l'école de Gif (1988)

[4] J. Faivre
Reconstruction and study of the multi-strange baryons in ultra-relativistic heavy-ion collisions at $\sqrt{S_{NN}} = 200$ *GeV with the* STAR *experiment at* RHIC.
Ph.D. thesis, Université Louis Pasteur, Strasbourg (2004)

[5] R. Fisher
The use of multiple measurements in taxonomic problems.
Ann.Eugen. **7**(1936):179–188

[6] J. Speltz : private communication

[7] J. Speltz, for the STAR collaboration
Investigating multi-strange baryon production in `Au-Au` *collisions at* $\sqrt{S_{NN}} = 62.4$ *GeV and its excitation function in high energy heavy ion collisions.*
J.Phys.G **31**(2005):S1025–S1028

Table of contents

Introduction **1**

1 Pattern classification **2**
1.1 General issue . . . 2
1.2 Problem of overtraining . . . 3
1.3 Estimation of the performance . . . 3

2 Observables : cuts variables **4**
2.1 Usual geometrical cuts . . . 4
2.2 Pointing angles . . . 6
2.3 Cosine of the decay angle . . . 6
2.4 Number of hits in the TPC . . . 7

3 Learning samples **7**

4 Principles of basic LDA **7**
4.1 Advantages of using LDA . . . 8
4.1.1 Improvement in statistics brought with respect to the classical cuts . . . 8
4.1.2 Easiness of cut tuning . . . 8
4.1.3 Cuts for particular conditions . . . 8
4.2 Fisher criterion . . . 8
4.3 Problems which show up . . . 12

5 LDA improvements **12**
5.1 Multicut-LDA . . . 12
5.2 Optimized criteria . . . 14
5.3 Algorithm of the optimized criterion I . . . 14
5.3.1 Function to maximize . . . 14
5.3.2 Maximization algorithm . . . 15
5.4 Determining the cut values . . . 17
5.5 Statistics necessary . . . 17
5.5.1 Evaluation of the statistics necessary . . . 17
5.5.2 Fisher criterion . . . 18
5.5.3 Optimized criterion . . . 20
5.5.4 Possible and impossible solutions . . . 20

6 How to choose the last LDA cut value **22**

7 Some results **24**
7.1 Candidates selected by both sets of cuts . . . 24
7.2 Characteristics of the LDA cuts . . . 24
7.3 Improvement brought . . . 26

Conclusions **29**

Bibliography **32**

List of figures

1 Pattern classification algorithm . . . 2
2 Performance of various boundaries on their training sample . . . 3
3 Geometry of a v0Vertex . . . 5
4 Geometry of a xiVertex . . . 5
5 Vectors used in the definition of the pointing angles . . . 6
6 Invariant mass distributions of the training samples . . . 7
7 Principle of basic LDA . . . 9
8 Analogy between the Fisher criterion and the resolution power of a telescope . . . 10
9 Case of a denominator tending towards zero in the Fisher criterion . . . 12
10 Multicut-LDA method . . . 13
11 Limit cone for the variation of the LDA vector . . . 16
12 Oscillations of the LDA vector with Fisher (1) . . . 18
13 Oscillations of the LDA vector with Fisher (2) . . . 19
14 Oscillations of the LDA vector with the optimized criterion . . . 19
15 Drawback of under-optimal LDA . . . 20
16 LDA performance as a function of the number of variables used . . . 21
17 Drawback of the PCA . . . 21
18 Performance diagrams of the cuts . . . 22
19 Method for tuning the last LDA cut . . . 23
20 Ω candidates selected by both sets of cuts . . . 24
21 Steepness factor of the LDA cuts on the Ξ signal . . . 25
22 Distributions of the steepness of the Ξ LDA cuts . . . 26
23 Distributions which lead to a null steepness of the LDA cut . . . 26
24 Exemplification of the calculation of the steepness factor of a cut . . . 27
25 Invariant mass distributions after the analysis cuts . . . 28
26 $\Omega + \overline{\Omega}$ invariant mass distributions for $1.4 < p_\perp < 1.8$ GeV/c . . . 28
27 LDA improvement factor . . . 29